

## STM32WB Bluetooth® Low Energy wireless interface

### Introduction

Bluetooth® Low Energy is a wireless personal area network technology designed and marketed by the Bluetooth special interest group (Bluetooth SIG), aimed at novel applications in the healthcare, fitness, beacons, security and home entertainment industries.

Bluetooth® Low Energy considerably reduces power consumption and cost compared to the standard Bluetooth, while maintaining a similar communication range.

Standard HCI commands are defined in the Bluetooth specification core V5.4, of which the Bluetooth® Low Energy specification is a part.

All proprietary commands are described in this application note.

## 1 General information

This document applies to STM32WB Series microcontrollers, based on Arm® cores.

**Note:** *Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.*



The parameter "size" in the tables of this document is expressed in bytes.

## 2 ACI/HCI commands

### 2.1 HCI commands

In Table 1 "Y" means that the corresponding command applies to the dedicated Bluetooth® Low Energy (BLE) stack variant, namely LO / BO / PO / BF / LB, respectively, for Link layer only / Beacon only / Peripheral only / Basic feature / Link layer only basic.

Table 1. HCI commands list

Command	OpCode	LO	PO	BO	BF	LB
HCI_DISCONNECT	0x0406	Y	Y	-	Y	Y
HCI_READ_REMOTE_VERSION_INFORMATION	0x041D	Y	Y	-	Y	Y
HCI_SET_EVENT_MASK	0x0C01	Y	Y	Y	Y	Y
HCI_RESET	0x0C03	Y	Y	Y	Y	Y
HCI_READ_TRANSMIT_POWER_LEVEL	0x0C2D	Y	Y	-	Y	Y
HCI_SET_CONTROLLER_TO_HOST_FLOW_CONTROL	0x0C31	Y	-	-	-	Y
HCI_HOST_BUFFER_SIZE	0x0C33	Y	-	-	-	Y
HCI_HOST_NUMBER_OF_COMPLETED_PACKETS	0x0C35	Y	-	-	-	Y
HCI_READ_LOCAL_VERSION_INFORMATION	0x1001	Y	Y	Y	Y	Y
HCI_READ_LOCAL_SUPPORTED_COMMANDS	0x1002	Y	-	Y	-	Y
HCI_READ_LOCAL_SUPPORTED_FEATURES	0x1003	Y	-	Y	-	Y
HCI_READ_BD_ADDR	0x1009	Y	Y	Y	Y	Y
HCI_READ_RSSI	0x1405	Y	Y	-	Y	Y
HCI_LE_SET_EVENT_MASK	0x2001	Y	Y	Y	Y	Y
HCI_LE_READ_BUFFER_SIZE	0x2002	Y	-	-	-	Y
HCI_LE_READ_LOCAL_SUPPORTED_FEATURES	0x2003	Y	-	Y	-	Y
HCI_LE_SET_RANDOM_ADDRESS	0x2005	Y	-	Y	-	Y
HCI_LE_SET_ADVERTISING_PARAMETERS	0x2006	Y	-	Y	-	Y
HCI_LE_READ_ADVERTISING_PHYSICAL_CHANNEL_TX_POWER	0x2007	Y	-	Y	-	Y
HCI_LE_SET_ADVERTISING_DATA	0x2008	Y	Y	Y	Y	Y
HCI_LE_SET_SCAN_RESPONSE_DATA	0x2009	Y	Y	Y	Y	Y
HCI_LE_SET_ADVERTISING_ENABLE	0x200A	Y	-	Y	-	Y
HCI_LE_SET_SCAN_PARAMETERS	0x200B	Y	-	Y	-	Y
HCI_LE_SET_SCAN_ENABLE	0x200C	Y	-	Y	-	Y
HCI_LE_CREATE_CONNECTION	0x200D	Y	-	-	-	Y
HCI_LE_CREATE_CONNECTION_CANCEL	0x200E	Y	-	-	-	Y
HCI_LE_READ_FILTER_ACCEPT_LIST_SIZE	0x200F	Y	-	Y	-	Y
HCI_LE_CLEAR_FILTER_ACCEPT_LIST	0x2010	Y	-	Y	-	Y
HCI_LE_ADD_DEVICE_TO_FILTER_ACCEPT_LIST	0x2011	Y	-	Y	-	Y
HCI_LE_REMOVE_DEVICE_FROM_FILTER_ACCEPT_LIST	0x2012	Y	-	Y	-	Y
HCI_LE_CONNECTION_UPDATE	0x2013	Y	-	-	Y	Y
HCI_LE_SET_HOST_CHANNEL_CLASSIFICATION	0x2014	Y	-	-	-	Y
HCI_LE_READ_CHANNEL_MAP	0x2015	Y	Y	-	Y	Y
HCI_LE_READ_REMOTE_FEATURES	0x2016	Y	Y	-	Y	Y

Command	OpCode	LO	PO	BO	BF	LB
HCI_LE_ENCRYPT	0x2017	Y	Y	-	Y	Y
HCI_LE RAND	0x2018	Y	Y	Y	Y	Y
HCI LE_ENABLE_ENCRYPTION	0x2019	Y	-	-	-	Y
HCI LE_LONG_TERM_KEY_REQUEST_REPLY	0x201A	Y	-	-	-	Y
HCI LE_LONG_TERM_KEY_REQUEST_NEGATIVE_REPLY	0x201B	Y	-	-	-	Y
HCI LE_READ_SUPPORTED_STATES	0x201C	Y	-	Y	-	Y
HCI LE_SET_DATA_LENGTH	0x2022	Y	Y	-	Y	Y
HCI LE_READ_SUGGESTED_DEFAULT_DATA_LENGTH	0x2023	Y	Y	-	Y	Y
HCI LE_WRITE_SUGGESTED_DEFAULT_DATA_LENGTH	0x2024	Y	Y	-	Y	Y
HCI LE_READ_LOCAL_P256_PUBLIC_KEY	0x2025	Y	Y	-	Y	Y
HCI LE_GENERATE_DHKEY	0x2026	Y	-	-	-	Y
HCI LE_ADD_DEVICE_TO_RESOLVING_LIST	0x2027	Y	-	-	-	Y
HCI LE_REMOVE_DEVICE_FROM_RESOLVING_LIST	0x2028	Y	-	-	-	Y
HCI LE_CLEAR_RESOLVING_LIST	0x2029	Y	-	-	-	Y
HCI LE_READ_RESOLVING_LIST_SIZE	0x202A	Y	-	-	-	Y
HCI LE_READ_PEER_RESOLVABLE_ADDRESS	0x202B	Y	Y	-	Y	Y
HCI LE_READ_LOCAL_RESOLVABLE_ADDRESS	0x202C	Y	Y	-	Y	Y
HCI LE_SET_ADDRESS_RESOLUTION_ENABLE	0x202D	Y	-	-	-	Y
HCI LE_SET_RESOLVABLE_PRIVATE_ADDRESS_TIMEOUT	0x202E	Y	Y	-	Y	Y
HCI LE_READ_MAXIMUM_DATA_LENGTH	0x202F	Y	Y	-	Y	Y
HCI LE_READ_PHY	0x2030	Y	-	-	Y	Y
HCI LE_SET_DEFAULT_PHY	0x2031	Y	-	-	Y	Y
HCI LE_SET_PHY	0x2032	Y	-	-	Y	Y
HCI LE_SET_ADVERTISING_SET_RANDOM_ADDRESS	0x2035	Y	-	-	-	Y
HCI LE_SET_EXTENDED_ADVERTISING_PARAMETERS	0x2036	Y	-	-	-	-
HCI LE_SET_EXTENDED_ADVERTISING_DATA	0x2037	Y	-	-	-	-
HCI LE_SET_EXTENDED_SCAN_RESPONSE_DATA	0x2038	Y	-	-	-	-
HCI LE_SET_EXTENDED_ADVERTISING_ENABLE	0x2039	Y	-	-	-	-
HCI LE_READ_MAXIMUM_ADVERTISING_DATA_LENGTH	0x203A	Y	-	-	-	-
HCI LE_READ_NUMBER_OF_SUPPORTED_ADVERTISING_SETS	0x203B	Y	-	-	-	-
HCI LE_REMOVE_ADVERTISING_SET	0x203C	Y	-	-	-	-
HCI LE_CLEAR_ADVERTISING_SETS	0x203D	Y	-	-	-	-
HCI LE_SET_EXTENDED_SCAN_PARAMETERS	0x2041	Y	-	-	-	-
HCI LE_SET_EXTENDED_SCAN_ENABLE	0x2042	Y	-	-	-	-
HCI LE_EXTENDED_CREATE_CONNECTION	0x2043	Y	-	-	-	-
HCI LE_READ_TRANSMIT_POWER	0x204B	Y	-	Y	-	Y
HCI LE_READ_RF_PATH_COMPENSATION	0x204C	Y	-	-	-	-
HCI LE_WRITE_RF_PATH_COMPENSATION	0x204D	Y	-	-	-	-
HCI LE_SET_PRIVACY_MODE	0x204E	Y	Y	-	Y	Y
HCI LE_GENERATE_DHKEY_V2	0x205E	Y	-	-	-	Y

## 2.1.1 HCI\_DISCONNECT

This command is used to terminate an existing connection. The Connection\_Handle parameter indicates the connection to be disconnected, and the Reason parameter indicates the reason for ending it. The remote controller receives the Reason parameter in the HCI\_DISCONNECTED\_COMPLETE\_EVENT. All synchronous connections on a physical link must be disconnected before the ACL connection on the same physical connection is disconnected. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.1.6].

**Table 2. HCI\_DISCONNECT input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle for which the command is given	0x0000 ... 0x0EFF
Reason	1	The reason for ending the connection	<ul style="list-style-type: none"><li>• 0x05: Authentication failure</li><li>• 0x13: Remote user terminated connection</li><li>• 0x14: Remote device terminated connection due to low resources</li><li>• 0x15: Remote device terminated connection due to power off</li><li>• 0x1A: Unsupported remote feature</li><li>• 0x3B: Unacceptable connection parameters</li></ul>

**Table 3. HCI\_DISCONNECT output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

### Events generated

- HCI\_COMMAND\_STATUS\_EVENT
- HCI\_DISCONNECTED\_COMPLETE\_EVENT

## 2.1.2 HCI\_READ\_REMOTE\_VERSION\_INFORMATION

This command obtains the version information values for the remote device identified by the Connection\_Handle parameter, which must be a Connection\_Handle for an ACL or LE connection. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.1.23].

**Table 4. HCI\_READ\_REMOTE\_VERSION\_INFORMATION input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the Connection_Handle version information to get	0x0000 ... 0x0EFF

**Table 5. HCI\_READ\_REMOTE\_VERSION\_INFORMATION output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

### Events generated

- HCI\_COMMAND\_STATUS\_EVENT
- HCI\_READ\_REMOTE\_VERSION\_INFORMATION\_COMPLETE\_EVENT

## 2.1.3 HCI\_SET\_EVENT\_MASK

This command is used to control the events generated by the HCI for the host. If the bit in the Event\_Mask is set to 1 the associated event is enabled. For an LE controller, the LE Meta event bit in the Event\_Mask enables or disables all LE events in the LE Meta event. The host must deal with each occurring event, the event mask allows it to control if it is interrupted. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.3.1].

**Table 6. HCI\_SET\_EVENT\_MASK input parameters**

Parameter	Size	Description	Possible values
Event_Mask	8	Event mask. Default: 0x2000FFFFFFFFFFFF	<p>Bitmask of:</p> <ul style="list-style-type: none"> <li>• 0x0000000000000000: No events specified</li> <li>• 0x0000000000000010: Disconnection complete event</li> <li>• 0x0000000000000080: Encryption change event</li> <li>• 0x00000000000000800: Read remote version information complete event</li> <li>• 0x000000000000008000: Hardware error event</li> <li>• 0x0000080000000000: Encryption key refresh complete event</li> <li>• 0x2000000000000000: LE Meta-event</li> </ul>

**Table 7. HCI\_SET\_EVENT\_MASK output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.1.4****HCI\_RESET**

This command resets the link layer on an LE controller, but does not affect the used HCI transport layer, as these layers can have their own reset mechanisms. After the reset is completed, the current operational state is lost, the controller enters standby mode, and automatically reverts to the default values for the parameters for which default values are defined in the specification. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.3.2].

**Note:**

*This command does not perform a hardware reset. The host does not send additional HCI commands before it receives the command Complete event related to the Reset command.*

**Input parameters:** none

**Table 8. HCI\_RESET output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.1.5****HCI\_READ\_TRANSMIT\_POWER\_LEVEL**

This command reads the values for the Transmit\_Power\_Level parameter for the Connection\_Handle specified for an ACL connection. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.3.35].

**Table 9. HCI\_READ\_TRANSMIT\_POWER\_LEVEL input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the Connection_Handle transmit power level setting to read	0x0000 ... 0x0EFF
Type	1	Current or maximum transmit power level	<ul style="list-style-type: none"> <li>• 0x00: Read current transmit power level</li> <li>• 0x01: Read maximum transmit power level</li> </ul>

**Table 10.** HCI\_READ\_TRANSMIT\_POWER\_LEVEL output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-
Connection_Handle	2	Connection handle related to the response	0x0000 ... 0x0EFF
Transmit_Power_Level	1	Size: 1 octet (signed integer), in dBm	-30 ... 20

**Events generated**

- HCI\_COMMAND\_COMPLETE\_EVENT

**2.1.6****HCI\_SET\_CONTROLLER\_TO\_HOST\_FLOW\_CONTROL**

This command is used by the host to turn flow control on or off for data and/or voice sent from the controller. If flow control is turned off, the host does not send the HCI\_HOST\_NUMBER\_OF\_COMPLETED\_PACKETS command: this command is ignored by the controller if it is sent when flow control is off.

If flow control is turned on/off for HCI ACL data packets and off/on for HCI synchronous data packets, HCI\_HOST\_NUMBER\_OF\_COMPLETED\_PACKETS commands sent by the host must contain only Connection\_Handles for ACL/synchronous connections.

If flow control is turned on for HCI ACL and HCI synchronous data packets, the host sends HCI\_HOST\_NUMBER\_OF\_COMPLETED\_PACKETS commands for both ACL and synchronous connections. If no connection exists, only the Flow\_Control\_Enable parameter is changed.

See Bluetooth spec. v5.4 [Vol 4, Part E, 7.3.38].

**Table 11.** HCI\_SET\_CONTROLLER\_TO\_HOST\_FLOW\_CONTROL input parameters

Parameter	Size	Description	Possible values
Flow_Control_Enable	1	Enable/Disable the flow control	<ul style="list-style-type: none"><li>• 0x00: Flow control off from controller to host (default).</li><li>• 0x01: Flow control on for HCI ACL data packets and off for HCI synchronous. Data packets from controller to host.</li><li>• 0x02: Flow control off for HCI ACL data packets and on for HCI synchronous. Data packets from controller to host.</li><li>• 0x03: Flow control on both for HCI ACL data packets and HCI synchronous. Data packets from controller to host.</li></ul>

**Table 12.** HCI\_SET\_CONTROLLER\_TO\_HOST\_FLOW\_CONTROL output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- HCI\_COMMAND\_COMPLETE\_EVENT

**2.1.7****HCI\_HOST\_BUFFER\_SIZE**

This command is used by the host to notify the controller about the maximum size of the data portion of HCI ACL and synchronous data packets sent from the controller. The controller segments the data to transmit according to their size, so that the HCI data packets contain data up to these sizes.

The command also notifies the controller about the total number of HCI ACL and synchronous data packets stored in the host data buffers. If flow control from the controller to the host is turned off, and the command has not been issued by the host, the controller sends HCI data packets with the length it wants to use. It is assumed that the host data buffer sizes are unlimited. If flow control from the controller to the host is turned on, the command, after a power-on or a reset, is always sent by the host before the first HCI\_HOST\_NUMBER\_OF\_COMPLETED\_PACKETS command.

The HCI\_SET\_CONTROLLER\_TO\_HOST\_FLOW\_CONTROL command is used to turn flow control on or off.

The Host\_ACL\_Data\_Packet\_Length parameter is used to determine the size of the L2CAP segments contained in ACL data packets, which are transferred from the controller to the host.

The Host\_Synchronous\_Data\_Packet\_Length parameter is used to determine the maximum size of HCI synchronous data packets. Both the host and the controller support command and event packets, where the data portion (excluding header) contained in the packets is 255 octets in size.

The Host\_Total\_Num\_ACL\_Data\_Packets parameter contains the total number of HCI ACL data packets stored in the data buffers of the host. The controller determines how the buffers must be divided between different Connection\_Handles.

The Host\_Total\_Num\_Synchronous\_Data\_Packets parameter gives the same information for HCI synchronous Data Packets.

See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.3.39].

Note:

*The Host\_ACL\_Data\_Packet\_Length and Host\_Synchronous\_Data\_Packet\_Length parameters do not include the length of the HCI Data Packet header.*

**Table 13. HCI\_HOST\_BUFFER\_SIZE input parameters**

Parameter	Size	Description	Possible values
Host_ACL_Data_Packet_Length	2	Maximum length (in octets) of the data portion of each HCI ACL data packet that the Host is able to accept. Must be greater than or equal to 251 bytes.	251 ... 65535
Host_Synchronous_Data_Packet_Length	1	Maximum length (in octets) of the data portion of each HCI synchronous data packet that the host is able to accept.	-
Host_Total_Num_ACL_Data_Packets	2	Total number of HCI ACL data packets that can be stored in the host data buffers.	1 ... 65535
Host_Total_Num_Synchronous_Data_Packets	2	Total number of HCI synchronous data packets that can be stored in the host data buffers.	-

**Table 14. HCI\_HOST\_BUFFER\_SIZE output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

#### Events generated

- HCI\_COMMAND\_COMPLETE\_EVENT

#### 2.1.8

#### HCI\_HOST\_NUMBER\_OF\_COMPLETED\_PACKETS

This command is used by the host to indicate to the controller the number of HCI data packets completed for each Connection\_Handle since the previous HCI\_HOST\_NUMBER\_OF\_COMPLETED\_PACKETS command was sent to the controller. This means that the corresponding buffer space has been freed.

Based on this information, the Host\_Total\_Num\_ACL\_Data\_Packets and Host\_Total\_Num\_Synchronous\_Data\_Packets parameters of the HCI\_HOST\_BUFFER\_SIZE command, the controller determines for which Connection\_Handles the next HCI data packets must be sent. The command is issued only if flow control from the controller to the host is on, and there is at least one connection, or if the controller is in local loop-back mode. Otherwise, the command is ignored. When the host has completed one or more HCI data packet(s), it sends a HCI\_HOST\_NUMBER\_OF\_COMPLETED\_PACKETS command to the controller, until it finally reports that all pending HCI data packets have been completed. The frequency at which this command is sent is manufacturer specific.

The HCI\_SET\_CONTROLLER\_TO\_HOST\_FLOW\_CONTROL command is used to turn flow control on or off. If flow control from the controller to the host is turned on, the HCI\_HOST\_BUFFER\_SIZE command is always sent by the host after a power-on or a reset before the first HCI\_HOST\_NUMBER\_OF\_COMPLETED\_PACKETS command is sent.

See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.3.40].

**Note:** This command is special, in the sense that no event is normally generated after the command has completed. The command can be sent at any time by the host when there is at least one connection, or if the controller is in local loop back mode, independently from other commands. The normal flow control is not used for this command.

**Table 15. HCI\_HOST\_NUMBER\_OF\_COMPLETED\_PACKETS input parameters**

Parameter	Size	Description	Possible values
Number_Of_Handles	1	Number of Connection_Handles and Host_Num_Of_Completed_Packets parameters pairs contained in this command.	0 - 255
Connection_Handle[i]	Number_Of_Handles * 2	Connection_Handle	0x0000-0x0EFF
Host_Num_Of_Completed_Packets[i]	Number_Of_Handles * 2	Number of HCI data packets completed for the associated Connection_Handle since the previous time the event was returned.	0x0000-0xFFFF

**Output parameters:** none

Unless masked away, no events are generated after this command has completed. However, if the command contains one or more invalid parameters, the controller returns a HCI\_COMMAND\_COMPLETE\_EVENT with a failure status indicating the invalid HCI command parameters error code. The host may send the command at any time when there is at least one connection, or if the controller is in local loop back mode. The normal flow control is not used for this command.

## 2.1.9 HCI\_READ\_LOCAL\_VERSION\_INFORMATION

This command reads the values of the version information for the local controller. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.4.1].

**Input parameters:** none

**Table 16. HCI\_READ\_LOCAL\_VERSION\_INFORMATION output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-
HCI_Version	1	Version of the HCI specification supported by the controller. See Bluetooth assigned numbers.	-
HCI_Subversion	2	Revision of the HCI implementation in the controller (vendor specific)	-
LMP_Version	1	Version of the current LMP supported by the controller. See Bluetooth assigned numbers.	-
Company_Identifier	2	Company identifier for the manufacturer of the controller. See Bluetooth assigned numbers.	-
LMP_Subversion	2	Subversion of the current LMP in the controller (implementation dependent)	-

**Events generated**

- HCI\_COMMAND\_COMPLETE\_EVENT

## 2.1.10 HCI\_READ\_LOCAL\_SUPPORTED\_COMMANDS

This command reads the list of HCI commands supported for the local controller, and returns the Supported\_Commands configuration parameter. If a command is supported, the feature underlying that command is also supported. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.4.2].

**Input parameters:** none

**Table 17. HCI\_READ\_LOCAL\_SUPPORTED\_COMMANDS output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-
Supported_Commands	64	Bit mask for each HCI command. If a bit is 1, the controller supports the corresponding command and the features required for it. Unsupported or undefined commands are set to 0.	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.1.11 HCI\_READ\_LOCAL\_SUPPORTED\_FEATURES**

This command requests a list of the supported features for the local controller, and returns a list of the LMP features. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.4.3].

**Input parameters:** none

**Table 18. HCI\_READ\_LOCAL\_SUPPORTED\_FEATURES output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-
LMP_Features	8	Bit mask list of LMP features	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.1.12 HCI\_READ\_BD\_ADDR**

On an LE controller, this command reads the public device address. If the controller does not have one, the value 0x000000000000 is returned. On an LE controller, the public address is the same as the BD\_ADDR. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.4.6].

**Input parameters:** none

**Table 19. HCI\_READ\_BD\_ADDR output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-
BD_ADDR	6	BD_ADDR (Bluetooth device address) of the device	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.1.13 HCI\_READ\_RSSI**

This command reads the received signal strength indication (RSSI) value from a controller. For an LE transport, a Connection\_Handle is used as the handle command parameter and return parameter. The meaning of the RSSI metric is an absolute receiver signal strength value in dBm, with  $\pm 6$  dB accuracy. If the RSSI cannot be read, the RSSI metric is set to 127. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.5.4].

**Table 20. HCI\_READ\_RSSI input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle for which the command is given	0x0000 ... 0x0EFF

**Table 21.** HCI\_READ\_RSSI output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-
Connection_Handle	2	Connection handle related to the response	0x0000 ... 0x0EFF
RSSI	1	N Size: 1 octet (signed integer) Units: dBm	<ul style="list-style-type: none"> <li>• 127: RSSI not available</li> <li>• -127 ... 20</li> </ul>

**Events generated**

HCI\_COMMAND\_COMPLETE\_EVENT

**2.1.14****HCI\_LE\_SET\_EVENT\_MASK**

This command is used to control which LE events are generated by the HCI for the host. If the bit in the LE\_Event\_Mask is set to 1, the associated event is enabled. The host must deal with each event generated by an LE controller. The event mask allows the host to control which events interrupt it. To generate an LE event, the LE Meta-Event bit in the Event\_Mask must be set. If not, LE events are not generated, regardless of how the LE\_Event\_Mask is set. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.1].

**Table 22.** HCI\_LE\_SET\_EVENT\_MASK input parameters

Parameter	Size	Description	Possible values
LE_Event_Mask	8	LE event mask. Default: 0x000000000003185F	<p>Bitmask of:</p> <ul style="list-style-type: none"> <li>• 0x0000000000000000: No LE events specified</li> <li>• 0x0000000000000001: LE connection complete event</li> <li>• 0x0000000000000002: LE advertising report event</li> <li>• 0x0000000000000004: LE connection update complete event</li> <li>• 0x0000000000000008: LE read remote used features complete event</li> <li>• 0x0000000000000010: LE long term key request event</li> <li>• 0x0000000000000020: LE remote connection parameter request event</li> <li>• 0x0000000000000040: ILE data length change event</li> <li>• 0x0000000000000080: LE read local P-256 public key complete event</li> <li>• 0x0000000000000100: LE generate DHKey complete event</li> <li>• 0x0000000000000200: LE enhanced connection complete event</li> <li>• 0x0000000000000400: LE direct advertising report event</li> <li>• 0x0000000000000800: LE PHY update complete event</li> <li>• 0x0000000000001000: LE extended advertising report event</li> <li>• 0x0000000000002000: LE periodic advertising sync established event</li> <li>• 0x0000000000004000: LE periodic advertising report event</li> <li>• 0x0000000000008000: LE periodic advertising sync lost event</li> <li>• 0x00000000000010000: LE extended scan timeout event</li> <li>• 0x00000000000020000: LE extended advertising set terminated event</li> <li>• 0x00000000000040000: LE scan request received event</li> <li>• 0x00000000000080000: LE channel selection algorithm event</li> </ul>

**Table 23.** HCI\_LE\_SET\_EVENT\_MASK output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

HCI\_COMMAND\_COMPLETE\_EVENT

## 2.1.15 HCI\_LE\_READ\_BUFFER\_SIZE

This command is used to read the maximum size of the data portion of HCI LE ACL data packets sent from the host to the controller. The host segments the transmitted data according to these values, so that the HCI data packets contain data with up to this size. The command also returns the total number of HCI LE ACL data packets stored in the data buffers of the controller. The command must be issued by the host before it sends any data to an LE controller. If the controller returns a length value of 0, the host uses the command to determine the size of the data buffers.

**Note:** Both the HCI\_READ\_BUFFER\_SIZE and HCI\_LE\_READ\_BUFFER\_SIZE commands can return non-0 values for the buffer length and number of packets parameters.

The HC\_LE\_ACL\_Data\_Packet\_Length return parameter is used to determine the size of the L2CAP PDU segments contained in ACL data packets, which are transferred from the host to the controller to be broken up into packets by the link layer. Both the host and the controller support command and event packets, where the data portion (excluding header) contained in the packets is 255 octets in size. The HC\_Total\_Num\_LE\_ACL\_Data\_Packets return parameter contains the total number of HCI ACL data packets that is stored in the data buffers of the controller. The host determines how the buffers are to be divided between different connection handles.

**Note:** The HC\_LE\_ACL\_Data\_Packet\_Length return parameter does not include the length of the HCI data packet header.

See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.2]

**Input parameters:** none

**Table 24. HCI\_LE\_READ\_BUFFER\_SIZE output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-
HC_LE_ACL_Data_Packet_Length	2	<ul style="list-style-type: none"><li>• 0x0000 No dedicated LE buffer, use Read_Buffer_Size command. 0x0001</li><li>• 0xFFFF Maximum length (in octets) of the data portion of each HCI ACL data packet that the controller is able to accept</li></ul>	-
HC_Total_Num_LE_ACL_Data_Packets	1	<ul style="list-style-type: none"><li>• 0x00 No dedicated LE buffer, use Read_Buffer_Size command</li><li>• 0x01 - 0xFF total number of HCI ACL data packets that is stored in the data buffers of the controller</li></ul>	-

### Events generated

HCI\_COMMAND\_COMPLETE\_EVENT

## 2.1.16 HCI\_LE\_READ\_LOCAL\_SUPPORTED\_FEATURES

This command requests the list of the supported LE features for the controller. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.3].

**Input parameters:** none

**Table 25. HCI\_LE\_READ\_LOCAL\_SUPPORTED\_FEATURES output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-
LE_Features	8	Bit mask list of LE features	-

### Events generated

HCI\_COMMAND\_COMPLETE\_EVENT

## 2.1.17 HCI\_LE\_SET\_RANDOM\_ADDRESS

This command is used by the host to set the LE random device address in the controller. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.4].

**Table 26. HCI\_LE\_SET\_RANDOM\_ADDRESS input parameters**

Parameter	Size	Description	Possible values
Random_Address	6	Random device address	-

**Table 27. HCI\_LE\_SET\_RANDOM\_ADDRESS output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

### Events generated

HCI\_COMMAND\_COMPLETE\_EVENT

## 2.1.18 HCI\_LE\_SET\_ADVERTISING\_PARAMETERS

This command is used by the host to set the advertising parameters. The Advertising\_Interval\_Min is lower than or equal to the Advertising\_Interval\_Max. The Advertising\_Interval\_Min and Advertising\_Interval\_Max cannot have the same value, to let the controller determine the best interval, considering other activities. For high duty cycle directed advertising, i.e. when Advertising\_Type is 0x01, the Advertising\_Interval\_Min and Advertising\_Interval\_Max parameters are ignored. The Advertising\_Type is used to determine the packet type used for advertising when enabled. The Advertising\_Interval\_Min and Advertising\_Interval\_Max are not set to less than 0x00A0 (100 ms) if the Advertising\_Type is set to 0x02 (ADV\_SCAN\_IND) or 0x03 (ADV\_NONCONN\_IND).

The Own\_Address\_Type determines if the advertising packets are identified with the public device address of the device, or a random device address as written by the HCI\_LE\_SET\_RANDOM\_ADDRESS command. If directed advertising is performed (Advertising\_Type is set to 0x01 or 0x04), the Direct\_Address\_Type and Direct\_Address are valid, otherwise they are ignored by the controller.

The Advertising\_Channel\_Map is a bit field that indicates the channels used when transmitting advertising packets. At least one channel bit is set in the Advertising\_Channel\_Map parameter. The Advertising\_Filter\_Policy parameter is ignored when directed advertising is enabled. The host does not issue this command when advertising is enabled in the controller; if it is, the command disallowed error code is used.

See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.5].

**Table 28. HCI\_LE\_SET\_ADVERTISING\_PARAMETERS input parameters**

Parameter	Size	Description	Possible values
Advertising_Interval_Min	2	Minimum advertising interval. Time = N * 0.625 ms	0x0020 (20 ms) ... 0x4000 (10240 ms)
Advertising_Interval_Max	2	Maximum advertising interval. Time = N * 0.625 ms	0x0020 (20 ms) ... 0x4000 (10240 ms)
Advertising_Type	1	Advertising type	<ul style="list-style-type: none"><li>• 0x00: ADV_IND (connectable undirected advertising)</li><li>• 0x01: ADV_DIRECT_IND, high duty cycle (connectable high duty cycle directed advertising)</li><li>• 0x02: ADV_SCAN_IND (scannable undirected advertising)</li><li>• 0x03: ADV_NONCONN_IND (non-connectable undirected advertising)</li><li>• 0x04: ADV_DIRECT_IND, low duty cycle (connectable low duty cycle directed advertising)</li></ul>
Own_Address_Type	1	Own address type. <ul style="list-style-type: none"><li>• 0x00: Public device address</li><li>• 0x01: Random device address</li></ul>	<ul style="list-style-type: none"><li>• 0x00: Public device address</li><li>• 0x01: Random device address</li></ul>

Parameter	Size	Description	Possible values
		<ul style="list-style-type: none"> <li>• 0x01: Random device address</li> <li>• 0x02: Controller generates resolvable private address based on the local IRK from resolving list. If resolving list contains no matching entry, use public address</li> <li>• 0x03: Controller generates resolvable private address based on the local IRK from resolving list. If resolving list contains no matching entry, use random address from LE_Set_Random_Address</li> </ul>	<ul style="list-style-type: none"> <li>• 0x02: Resolvable private or public address</li> <li>• 0x03: Resolvable private or random address</li> </ul>
Peer_Address_Type	1	The address type of the peer device	<ul style="list-style-type: none"> <li>• 0x00: Public device address</li> <li>• 0x01: Random device address</li> </ul>
Peer_Address	6	Public device address, random device address, public identity address or random (static) identity address of the device to be connected	-
Advertising_Channel_Map	1	Advertising channel map. Default: 00000111b (all channels enabled).	Bitmask of: <ul style="list-style-type: none"> <li>• 0x01: ch 37</li> <li>• 0x02: ch 38</li> <li>• 0x04: ch 39</li> </ul>
Advertising_Filter_Policy	1	Advertising filter policy	<ul style="list-style-type: none"> <li>• 0x00: Allow scan request from any, allow connect request from any</li> <li>• 0x01: Allow scan request from white list only, allow connect request from any</li> <li>• 0x02: Allow scan request from any, allow connect request from white list only</li> <li>• 0x03: Allow scan request from white list only, allow connect request from white list only</li> </ul>

**Table 29. HCI\_LE\_SET\_ADVERTISING\_PARAMETERS output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**
**HCI\_COMMAND\_COMPLETE\_EVENT**
**2.1.19**
**HCI\_LE\_READ\_ADVERTISING\_PHYSICAL\_CHANNEL\_TX\_POWER**

This command is used by the host to read the transmit power level used for LE advertising channel packets. See Bluetooth spec. v5.4 [Vol 4, Part E, 7.8.6].

**Input parameters:** none

**Table 30. HCI\_LE\_READ\_ADVERTISING\_PHYSICAL\_CHANNEL\_TX\_POWER output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-
Transmit_Power_Level	1	Size: 1 octet (signed integer) Units: dBm Accuracy: ± 4 dBm	-20 ... 10

**Events generated**  
`HCI_COMMAND_COMPLETE_EVENT`

#### 2.1.20 **HCI\_LE\_SET\_ADVERTISING\_DATA**

This command sets the data used in advertising packets with a data field. Only the significant part of the Advertising\_Data is transmitted in the packets. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.7].

**Table 31. HCI\_LE\_SET\_ADVERTISING\_DATA input parameters**

Parameter	Size	Description	Possible values
Advertising_Data_Length	1	The number of significant octets in the following data field	-
Advertising_Data	31	31 octets of data formatted	-

**Table 32. HCI\_LE\_SET\_ADVERTISING\_DATA output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**  
`HCI_COMMAND_COMPLETE_EVENT`

#### 2.1.21 **HCI\_LE\_SET\_SCAN\_RESPONSE\_DATA**

This command is used to provide data used in scanning packets with a data field. Only the significant part of the Scan\_Response\_Data is transmitted in the packets. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.8]

**Table 33. HCI\_LE\_SET\_SCAN\_RESPONSE\_DATA input parameters**

Parameter	Size	Description	Possible values
Scan_Response_Data_Length	1	The number of significant octets in the following data field	-
Scan_Response_Data	31	31 octets of data formatted	-

**Table 34. HCI\_LE\_SET\_SCAN\_RESPONSE\_DATA output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**  
• `HCI_COMMAND_COMPLETE_EVENT`

#### 2.1.22 **HCI\_LE\_SET\_ADVERTISING\_ENABLE**

This command is used to request the controller to start or stop advertising. The controller manages the timing of advertisements according to the parameters given in the `HCI_LE_SET_ADVERTISING_PARAMETERS` command. The controller continues advertising until the host issues an `HCI_LE_SET_ADVERTISING_ENABLE` command with Advertising\_Enable set to 0x00 (advertising is disabled), or until a connection is created, or until the advertising is timed out due to high duty cycle directed advertising. In these cases, advertising is disabled. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.9].

**Table 35. HCI\_LE\_SET\_ADVERTISING\_ENABLE input parameters**

Parameter	Size	Description	Possible values
Advertising_Enable	1	Enable/disable advertising. Default is 0 (disabled).	<ul style="list-style-type: none"><li>• 0x00: Advertising is disabled</li><li>• 0x01: Advertising is enabled</li></ul>

**Table 36. HCI\_LE\_SET\_ADVERTISING\_ENABLE output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)
- [HCI\\_LE\\_CONNECTION\\_COMPLETE\\_EVENT](#)

**2.1.23****HCI\_LE\_SET\_SCAN\_PARAMETERS**

This command is used to set the scan parameters. The LE\_Scan\_Type parameter controls the type of scan to perform. The LE\_Scan\_Interval and LE\_Scan\_Window parameters are recommendations from the host on how long (LE\_Scan\_Window) and how frequently (LE\_Scan\_Interval) the controller must scan.

The LE\_Scan\_Window parameter is always set to a value smaller than or equal to the value set for the LE\_Scan\_Interval parameter. If they are set to the same value, scanning runs continuously. The Own\_Address\_Type parameter determines the address (public or random device) used when performing active scan. The host does not issue this command when scanning is enabled in the controller; if it is, the command disallowed error code is used.

See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.10].

**Table 37. HCI\_LE\_SET\_SCAN\_PARAMETERS input parameters**

Parameter	Size	Description	Possible values
LE_Scan_Type	1	Passive or active scanning. With active scanning SCAN_REQ packets are sent.	<ul style="list-style-type: none"> <li>• 0x00: Passive scanning</li> <li>• 0x01: Active scanning</li> </ul>
LE_Scan_Interval	2	Time interval from when the controller started its last LE scan until it begins the next one.  Time = N * 0.625 ms	<ul style="list-style-type: none"> <li>• 0x0004 (2.5 ms) ... 0x4000 (10240.0 ms)</li> </ul>
LE_Scan_Window	2	Amount of time for the duration of the LE scan. LE_Scan_Window is less than or equal to LE_Scan_Interval.  Time = N * 0.625 ms	<ul style="list-style-type: none"> <li>• 0x0004 (2.5 ms) ... 0x4000 (10240.0 ms)</li> </ul>
Own_Address_Type	1	Own address type. <ul style="list-style-type: none"> <li>• 0x00: Public device address</li> <li>• 0x01: Random device address</li> <li>• 0x02: Controller generates resolvable private address based on the local IRK from resolving list. If resolving list contains no matching entry, use public address.</li> <li>• 0x03: Controller generates resolvable private address based on the local IRK from resolving list. If resolving list contains no matching entry, use random address from LE_Set_Random_Address</li> </ul>	<ul style="list-style-type: none"> <li>• 0x00: Public device address</li> <li>• 0x01: Random device address</li> <li>• 0x02: Resolvable private address or public address</li> <li>• 0x03: Resolvable private address or random address</li> </ul>
Scanning_Filter_Policy	1	<ul style="list-style-type: none"> <li>• 0x00: accepts all advertisement packets. Directed advertising packets which are not addressed for this device is ignored.</li> <li>• 0x01: ignores advertisement packets from devices not in the white list. Directed advertising packets not addressed to this device are ignored</li> <li>• 0x02: accepts all undirected advertisement packets. Directed advertisement packets where initiator address is a RPA and directed advertisement packets addressed to this device are accepted.</li> <li>• 0x03: accept all undirected advertisement packets from devices that are in the white List. Directed advertisement packets where initiator address is RPA and directed advertisement packets addressed to this device are accepted.</li> </ul>	<ul style="list-style-type: none"> <li>• 0x00: Accept all</li> <li>• 0x01: Ignore devices not in the white list</li> <li>• 0x02: Accept all (use resolving list)</li> <li>• 0x03: Ignore devices not in the white list (use resolving list)</li> </ul>

**Table 38. HCI\_LE\_SET\_SCAN\_PARAMETERS output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**[HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)**2.1.24****HCI\_LE\_SET\_SCAN\_ENABLE**

This command is used to start scanning, to discover advertising devices nearby. The Filter\_Duplicates parameter controls whether the link layer filters duplicate advertising reports to the host, or if the link layer must generate advertising reports for each received packet. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.11].

**Table 39. HCI\_LE\_SET\_SCAN\_ENABLE input parameters**

Parameter	Size	Description	Possible values
LE_Scan_Enable	1	Enable/disable scan. Default is 0 (disabled)	<ul style="list-style-type: none"> <li>• 0x00: Scanning disabled</li> <li>• 0x01: Scanning enabled</li> </ul>
Filter_Duplicates	1	Enable/disable duplicate filtering	<ul style="list-style-type: none"> <li>• 0x00: Duplicate filtering disabled</li> <li>• 0x01: Duplicate filtering enabled</li> </ul>

**Table 40. HCI\_LE\_SET\_SCAN\_ENABLE output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)
- [HCI\\_LE\\_ADVERTISING\\_REPORT\\_EVENT](#)

**2.1.25****HCI\_LE\_CREATE\_CONNECTION**

This command is used to create a link layer connection to a connectable advertiser.

The LE\_Scan\_Interval and LE\_Scan\_Window parameters are recommendations from the host on how long (LE\_Scan\_Window) and how frequently (LE\_Scan\_Interval) the controller must scan. The LE\_Scan\_Window parameter is set to a value smaller than or equal to the value set for the LE\_Scan\_Interval parameter. If both are set to the same value, scanning runs continuously.

The Initiator\_Filter\_Policy is used to determine whether the white list is used. If not used, the Peer\_Address\_Type and the Peer\_Address parameters specify the type and the address of the advertising device to connect to. The link layer sets the address in the CONNECT\_REQ packets to either the public device address, or the random device addressed, based on the Own\_Address\_Type parameter.

The Conn\_Interval\_Min and Conn\_Interval\_Max parameters define the minimum and maximum allowed connection interval. The Conn\_Interval\_Min parameter is not greater than the Conn\_Interval\_Max parameter. The Conn\_Latency parameter defines the maximum allowed connection latency.

The Supervision\_Timeout parameter defines the link supervision timeout for the connection. The Supervision\_Timeout (in ms) is larger than  $(1 + \text{Conn_Latency}) * \text{Conn_Interval_Max} * 2$  (Conn\_Interval\_Max is given in ms). The Minimum\_CE\_Length and Maximum\_CE\_Length parameters are informative parameters providing the controller with the expected minimum and maximum length of the connection events. The Minimum\_CE\_Length parameter is less than or equal to the Maximum\_CE\_Length parameter.

The host does not issue this command when another LE\_Create\_Connection is pending in the controller; if this occurs, the controller returns the command disallowed error code.

See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.12].

**Table 41. HCI\_LE\_CREATE\_CONNECTION input parameters**

Parameter	Size	Description	Possible values
LE_Scan_Interval	2	Time from when the controller started its last LE scan until it begins the subsequent LE scan.  Time = N * 0.625 ms	0x0004 (2.5 ms) ... 0x4000 (10240.0 ms)
LE_Scan_Window	2	Amount of time for the duration of the LE scan. LE_Scan_Window is less than or equal to LE_Scan_Interval.  Time = N * 0.625 ms	0x0004 (2.5 ms) ... 0x4000 (10240.0 ms)
Initiator_Filter_Policy	1	<ul style="list-style-type: none"> <li>• 0x00: White list is not used to determine which advertiser to connect to. Peer_Address_Type and Peer_Address is used</li> <li>• 0x01: White list is used to determine which advertiser to connect to. Peer_Address_Type and Peer_Address is ignored</li> </ul>	<ul style="list-style-type: none"> <li>• 0x00: White list not used</li> <li>• 0x01: White list used</li> </ul>
Peer_Address_Type	1	<ul style="list-style-type: none"> <li>• 0x00: Public device address</li> <li>• 0x01: Random device address</li> <li>• 0x02: Public identity address (corresponds to resolved private address)</li> <li>• 0x03: Random (static) identity address (corresponds to resolved private address)</li> </ul>	<ul style="list-style-type: none"> <li>• 0x00: Public device address</li> <li>• 0x01: Random device address</li> <li>• 0x02: Public identity address</li> <li>• 0x03: Random (static) identity address</li> </ul>
Peer_Address	1	Public device address or random device address of the device to be connected.	-
Own_Address_Type	1	Own address type. <ul style="list-style-type: none"> <li>• 0x00: Public device address</li> <li>• 0x01 Random device address</li> <li>• 0x02: Controller generates resolvable private address based on the local IRK from resolving list. If resolving list contains no matching entry, use public address.</li> <li>• 0x03: Controller generates resolvable private address based on the local IRK from resolving list. If resolving list contains no matching entry, use random address from LE_Set_Random_Address.</li> </ul>	<ul style="list-style-type: none"> <li>• 0x00: Public device address</li> <li>• 0x01: Random device address</li> <li>• 0x02: Resolvable private address or public address</li> <li>• 0x03: Resolvable private address or random address</li> </ul>
Conn_Interval_Min	2	Minimum value for the connection event interval. This is less than or equal to Conn_Interval_Max.  Time = N * 1.25 ms	0x0006 (7.50 ms) ... 0x0C80 (4000.00 ms)
Conn_Interval_Max	2	Maximum value for the connection event interval. This is greater than or equal to Conn_Interval_Min.  Time = N * 1.25 ms	0x0006 (7.50 ms) ... 0x0C80 (4000.00 ms)
Conn_Latency	2	Peripheral latency for the connection, in number of connection events	0x0000 ... 0x01F3
Supervision_Timeout	2	Supervision timeout for the LE Link. It is a multiple of 10 ms, and larger than (1 + connPeripheralLatency) * connInterval * 2.  Time = N * 10 ms	0x000A (100 ms) ... 0x0C80 (32000 ms)
Minimum_CE_Length	2	Information parameter about the minimum length of connection needed for this LE connection.  Time = N * 0.625 ms	0x0000 (0.000 ms) ... 0xFFFF (40959.375 ms)
Maximum_CE_Length	2	Information parameter about the maximum length of connection needed for this LE connection.  Time = N * 0.625 ms	0x0000 (0.000 ms) ... 0xFFFF (40959.375 ms)

**Table 42. HCI\_LE\_CREATE\_CONNECTION output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_STATUS\\_EVENT](#)
- [HCI\\_LE\\_CONNECTION\\_COMPLETE\\_EVENT](#)
- [HCI\\_LE\\_ENHANCED\\_CONNECTION\\_COMPLETE\\_EVENT](#)

**2.1.26****HCI\_LE\_CREATE\_CONNECTION\_CANCEL**

This command is used to cancel the HCI\_LE\_CREATE\_CONNECTION command, hence it issued only after it, a command status event has been received, and before the HCI\_LE\_CONNECTION\_COMPLETE\_EVENT. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.13].

**Input parameters:** none

**Table 43. HCI\_LE\_CREATE\_CONNECTION\_CANCEL output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)
- [HCI\\_LE\\_CONNECTION\\_COMPLETE\\_EVENT](#)
- [HCI\\_LE\\_ENHANCED\\_CONNECTION\\_COMPLETE\\_EVENT](#)

**2.1.27****HCI\_LE\_READ\_FILTER\_ACCEPT\_LIST\_SIZE**

This command is used to read the total number of white list entries that can be stored in the controller. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.14].

**Input parameters:** none

**Table 44. HCI\_LE\_READ\_FILTER\_ACCEPT\_LIST\_SIZE output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-
White_List_Size	1	Total number of white list entries that can be stored in the controller	-

**Events generated**

[HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.1.28****HCI\_LE\_CLEAR\_FILTER\_ACCEPT\_LIST**

This command is used to clear the white list stored in the controller. It can be used at any time except when:

- the advertising filter policy uses the white list and advertising is enabled
- the scanning filter policy uses the white list and scanning is enabled
- the initiator filter policy uses the white list and an LE\_Create\_Connection command is outstanding

See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.15].

**Input parameters:** none

**Table 45.** HCI\_LE\_CLEAR\_FILTER\_ACCEPT\_LIST output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**[HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)**2.1.29****HCI\_LE\_ADD\_DEVICE\_TO\_FILTER\_ACCEPT\_LIST**

This command is used to add a single device to the white list stored in the controller. It can be used at any time, except when:

- the advertising filter policy uses the white list and advertising is enabled
- the scanning filter policy uses the white list and scanning is enabled
- the initiator filter policy uses the white list and a create connection command is outstanding

See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.16].

**Table 46.** HCI\_LE\_ADD\_DEVICE\_TO\_FILTER\_ACCEPT\_LIST input parameters

Parameter	Size	Description	Possible values
Address_Type	1	Address type	<ul style="list-style-type: none"><li>0x00: Public device address</li><li>0x01: Random device address</li></ul>
Address	6	Public device address or random device address of the device to be added to the white list.	-

**Table 47.** HCI\_LE\_ADD\_DEVICE\_TO\_FILTER\_ACCEPT\_LIST output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**[HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)**2.1.30****HCI\_LE\_REMOVE\_DEVICE\_FROM\_FILTER\_ACCEPT\_LIST**

This command is used to remove a single device from the white list stored in the controller. It can be used at any time, except when:

- the advertising filter policy uses the white list and advertising is enabled
- the scanning filter policy uses the white list and scanning is enabled
- the initiator filter policy uses the white list and a create connection command is outstanding

See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.17].

**Table 48.** HCI\_LE\_REMOVE\_DEVICE\_FROM\_FILTER\_ACCEPT\_LIST input parameters

Parameter	Size	Description	Possible values
Address_Type	1	Address type	<ul style="list-style-type: none"><li>0x00: Public device address</li><li>0x01: Random device address</li></ul>
Address	6	Public device address or random device address of the device to be removed from the white list	-

**Table 49.** HCI\_LE\_REMOVE\_DEVICE\_FROM\_FILTER\_ACCEPT\_LIST output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated****HCI\_COMMAND\_COMPLETE\_EVENT****2.1.31 HCI\_LE\_CONNECTION\_UPDATE**

This command, supported only on central side, is used to change the link layer parameters of a connection:

- The Conn\_Interval\_Min and Conn\_Interval\_Max parameters are used to define the minimum and maximum allowed connection interval. The Conn\_Interval\_Min parameter is not greater than the Conn\_Interval\_Max parameter.
- The Conn\_Latency parameter defines the maximum allowed connection latency. The Supervision\_Timeout parameter defines the link supervision timeout for the LE link.
- The Supervision\_Timeout, in ms, is larger than  $(1 + Conn_Latency) * Conn_Interval_Max$  (in ms) \* 2.
- The Minimum\_CE\_Length and Maximum\_CE\_Length provide the controller information about the expected minimum and maximum length of the connection events. The Minimum\_CE\_Length is less than or equal to the Maximum\_CE\_Length.

The actual parameter values selected by the link layer may be different from the parameter values provided by the host through this command.

See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.18].

**Table 50. HCI\_LE\_CONNECTION\_UPDATE input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle for which the command is given	0x0000 ... 0x0EFF
Conn_Interval_Min	2	Minimum value for the connection event interval, less than or equal to Conn_Interval_Max. Time = N * 1.25 ms	0x0006 (7.5 ms) ... 0x0C80 (4000.0 ms)
Conn_Interval_Max	2	Maximum value for the connection event interval, greater than or equal to Conn_Interval_Min. Time = N * 1.25 ms	0x0006 (7.5 ms) ... 0x0C80 (4000.0 ms)
Conn_Latency	2	Peripheral latency for the connection, in number of connection events	0x0000 ... 0x01F3
Supervision_Timeout	2	Supervision timeout for the LE link. It is a multiple of 10 ms, and larger than $(1 + connPeripheralLatency) * connInterval * 2$ . Time = N * 10 ms	0x000A (100 ms) ... 0x0C80 (32000 ms)
Minimum_CE_Length	2	Information parameter about the minimum length of connection needed for this LE connection. Time = N * 0.625 ms	0x0000 (0.000 ms) ... 0xFFFF (40959.375 ms)
Maximum_CE_Length	2	Information parameter about the maximum length of connection needed for this LE connection. Time = N * 0.625 ms	0x0000 (0.000 ms) ... 0xFFFF (40959.375 ms)

**Table 51. HCI\_LE\_CONNECTION\_UPDATE output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- HCI\_COMMAND\_STATUS\_EVENT**
- HCI\_LE\_CONNECTION\_COMPLETE\_EVENT**

## 2.1.32

**HCI\_LE\_SET\_HOST\_CHANNEL\_CLASSIFICATION**

This command, used only when the local device supports the central role, allows the host to specify a channel classification for data channels based on its local information. The classification persists until overwritten with a subsequent command, or until the controller is reset using the Reset command. If this command is used, the host must send it within 10 s after it has known that the channel classification has changed. The interval between two sent commands must be at least 1 s. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.19].

**Table 52. HCI\_LE\_SET\_HOST\_CHANNEL\_CLASSIFICATION input parameters**

Parameter	Size	Description	Possible values
LE_Channel_Map	5	Contains 37 1-bit fields, the $n^{\text{th}}$ field (range 0 to 36) contains the value for the link layer channel index $n$ . Channel $n$ bad = 0, Channel $n$ unknown = 1. The most significant bits are reserved and set to 0. At least one channel is marked as unknown.	-

**Table 53. HCI\_LE\_SET\_HOST\_CHANNEL\_CLASSIFICATION output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

HCI\_COMMAND\_COMPLETE\_EVENT

## 2.1.33

**HCI\_LE\_READ\_CHANNEL\_MAP****Description**

This command returns the current Channel\_Map for the specified Connection\_Handle. The returned value indicates the state of the Channel\_Map specified by the last transmitted or received Channel\_Map (in a CONNECT\_REQ or LL\_CHANNEL\_MAP\_REQ message) for the specified Connection\_Handle, regardless of whether the central has received or not an acknowledgement. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.20].

**Table 54. HCI\_LE\_READ\_CHANNEL\_MAP input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle for which the command is given	0x0000 ... 0x0EFF

**Table 55. HCI\_LE\_READ\_CHANNEL\_MAP output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-
Connection_Handle	2	Connection handle related to the response	0x0000 ... 0x0EFF
LE_Channel_Map	5	This parameter contains 37 1-bit fields. The $n^{\text{th}}$ such field (in the range 0 to 36) contains the value for the link layer channel index $n$ . Channel $n$ unused = 0. Channel $n$ used = 1. The most significant bits are reserved, and set to 0.	-

**Events generated**

HCI\_COMMAND\_COMPLETE\_EVENT

## 2.1.34

**HCI\_LE\_READ\_REMOTE\_FEATURES**

This command requests a list of the used LE features from the remote device, and returns a list of the used LE features. It can be issued on both the client and the server. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.21].

**Table 56. HCI\_LE\_READ\_REMOTE\_FEATURES input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle for which the command is given	0x0000 ... 0x0EFF

**Table 57. HCI\_LE\_READ\_REMOTE\_FEATURES output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)
- [HCI\\_LE\\_READ\\_REMOTE\\_FEATURES\\_COMPLETE\\_EVENT](#)

### 2.1.35 [HCI\\_LE\\_ENCRYPT](#)

This command is used to request the controller to encrypt the Plaintext\_Data using the key provided by the command, and returns the Encrypted\_Data to the host. The AES-128 bit block cipher is defined in NIST Publication FIPS-197 (<http://csrc.nist.gov>). See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.22].

**Table 58. HCI\_LE\_ENCRYPT input parameters**

Parameter	Size	Description	Possible values
Key	16	128-bit key for the encryption of data given in the command	-
Plaintext_Data	16	128-bit data block to encrypt	-

**Table 59. HCI\_LE\_ENCRYPT output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-
Encrypted_Data	16	128-bit encrypted data block	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

### 2.1.36 [HCI\\_LE\\_RAND](#)

This command is used to request the controller to generate eight octets of random data to be sent to the host. Random\_Number is generated according to [Vol 2] Part H, Section 2 if the LE feature (LL encryption) is supported. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.23].

**Input parameters:** none

**Table 60. HCI\_LE\_RAND output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-
Random_Number	8	Random number	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

### 2.1.37 HCI\_LE\_ENABLE\_ENCRYPTION

This command is used to authenticate the encryption key associated with the remote device specified by the connection handle, and, once authenticated, encrypts the connection. If the connection is already encrypted the controller pauses before attempting to authenticate the given encryption key, and then re-encrypts the connection. While encryption is paused, no user data are transmitted. On an authentication failure, the connection is automatically disconnected by the link layer. If this command succeeds, the connection is encrypted. This command is used only when the local device role is central. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.24].

**Table 61. HCI\_LE\_ENABLE\_ENCRYPTION input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle for which the command is given	0x0000 ... 0x0EFF
Random_Number	8	64-bit random number	-
Encrypted_Diversifier	2	16-bit encrypted diversifier	-
Long_Term_Key	16	128-bit long term key	-

**Table 62. HCI\_LE\_ENABLE\_ENCRYPTION output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

#### Events generated

- `HCI_COMMAND_STATUS_EVENT`
- `HCI_ENCRYPTION_CHANGE_EVENT`
- `HCI_ENCRYPTION_KEY_REFRESH_COMPLETE_EVENT`

### 2.1.38 HCI\_LE\_LONG\_TERM\_KEY\_REQUEST\_REPLY

This command is used to reply to an LE long term key request event from the controller, and specifies the Long\_Term\_Key parameter used for this Connection\_Handle. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.25].

**Table 63. HCI\_LE\_LONG\_TERM\_KEY\_REQUEST\_REPLY input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle for which the command is given	0x0000 ... 0x0EFF
Long_Term_Key	16	128-bit long term key	-

**Table 64. HCI\_LE\_LONG\_TERM\_KEY\_REQUEST\_REPLY output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-
Connection_Handle	2	Connection handle related to the response	0x0000 ... 0x0EFF

#### Events generated

- `HCI_COMMAND_COMPLETE_EVENT`

### 2.1.39 HCI\_LE\_LONG\_TERM\_KEY\_REQUEST\_NEGATIVE\_REPLY

This command is used to reply to an LE long term key request event from the controller if the host cannot provide a long term key for this Connection\_Handle. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.26].

**Table 65.** HCI\_LE\_LONG\_TERM\_KEY\_REQUEST\_NEGATIVE\_REPLY input parameters

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle for which the command is given	0x0000 ... 0x0EFF

**Table 66.** HCI\_LE\_LONG\_TERM\_KEY\_REQUEST\_NEGATIVE\_REPLY output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-
Connection_Handle	2	Connection handle related to the response	0x0000 ... 0x0EFF

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.1.40****HCI\_LE\_READ\_SUPPORTED\_STATES**

This command reads the states and state combinations that the link layer supports. LE\_States is an 8-octet bit field. If a bit is set to 1, this state or state combination is supported by the controller. Multiple bits in LE\_States can be set to 1, to indicate support for multiple state and state combinations.

The advertising type and the initial state combinations are set only if the corresponding advertising types and central role combination are set. The scanning types and the initial state combinations are set only if the corresponding scanning types and central role combination are set.

See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.27].

**Input parameters:** none

**Table 67.** HCI\_LE\_READ\_SUPPORTED\_STATES output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-
LE_States	8	State or state combination is supported by the controller	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.1.41****HCI\_LE\_SET\_DATA\_LENGTH**

This command allows the host to suggest maximum transmission packet size and maximum packet transmission time (connMaxTxOctets and connMaxTxTime) for a given connection. The controller can use different values, based on local information. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.33].

**Table 68.** HCI\_LE\_SET\_DATA\_LENGTH input parameters

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle for which the command is given	0x0000 ... 0x0EFF
TxOctets	2	Preferred maximum number of payload octets that the local controller includes in a single link layer packet on this connection	0x001B ... 0x00FB
TxTime	2	Preferred maximum number of microseconds that the local controller must use to transmit a single link layer packet on this connection	0x0148 ... 0x4290

**Table 69.** HCI\_LE\_SET\_DATA\_LENGTH output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.1.42****HCI\_LE\_READ\_SUGGESTED\_DEFAULT\_DATA\_LENGTH**

With this command the host reads the suggested values (SuggestedMaxTxOctets and SuggestedMaxTxTime) for the controller maximum transmitted number of payload octets and maximum packet transmission time, for new connections. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.34].

**Input parameters:** none

**Table 70. HCI\_LE\_READ\_SUGGESTED\_DEFAULT\_DATA\_LENGTH output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-
SuggestedMaxTxOctets	2	The host suggested value for the controller maximum transmitted number of payload octets, to use for new connections	0x001B ... 0x00FB
SuggestedMaxTxTime	2	The host suggested value for the controller maximum packet transmission time, to use for new connections	0x0148 ... 0x4290

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.1.43****HCI\_LE\_WRITE\_SUGGESTED\_DEFAULT\_DATA\_LENGTH**

With this command the host indicates to the controller values for the maximum transmission number of payload octets and maximum packet transmission time, for new connections. The controller can use different values for connInitialMaxTxOctets and connInitialMaxTxTime, based on local information. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.35].

**Table 71. HCI\_LE\_WRITE\_SUGGESTED\_DEFAULT\_DATA\_LENGTH input parameters**

Parameter	Size	Description	Possible values
SuggestedMaxTxOctets	2	Suggested value for the controller maximum transmitted number of payload octets, to use for new connections	0x001B ... 0x00FB
SuggestedMaxTxTime	2	The host suggested value for the controller maximum packet transmission time, to use for new connections	0x0148 ... 0x4290

**Table 72. HCI\_LE\_WRITE\_SUGGESTED\_DEFAULT\_DATA\_LENGTH output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.1.44****HCI\_LE\_READ\_LOCAL\_P256\_PUBLIC\_KEY**

This command is used to return the local P-256 public key from the controller, which generates a new P- 256 public/private key pair upon receipt of the command. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.36].

**Input parameters:** none

**Table 73. HCI\_LE\_READ\_LOCAL\_P256\_PUBLIC\_KEY output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_STATUS\\_EVENT](#)
- [HCI\\_LE\\_READ\\_LOCAL\\_P256\\_PUBLIC\\_KEY\\_COMPLETE\\_EVENT](#)

**2.1.45****HCI\_LE\_GENERATE\_DHKEY**

This command is used to generate a Diffie-Hellman key in the controller for use over the LE transport, taking the remote P-256 public key as input. The generation uses the private key from the [HCI\\_LE\\_READ\\_LOCAL\\_P256\\_PUBLIC\\_KEY](#) command. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.37].

**Table 74. HCI\_LE\_GENERATE\_DHKEY input parameters**

Parameter	Size	Description	Possible values
Remote_P256_Public_Key	64	Remote P-256 public key: X, Y format octets <ul style="list-style-type: none"><li>• 31:0: X coordinate octets</li><li>• 63:32: Y coordinate little endian format</li></ul>	-

**Table 75. HCI\_LE\_GENERATE\_DHKEY output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_STATUS\\_EVENT](#)
- [HCI\\_LE\\_GENERATE\\_DHKEY\\_COMPLETE\\_EVENT](#)

**2.1.46****HCI\_LE\_ADD\_DEVICE\_TO\_RESOLVING\_LIST**

This command adds a device to the list of address translations used to resolve resolvable private addresses in the controller. It cannot be used when address translation is enabled in the controller and:

- advertising is enabled
- scanning is enabled
- [HCI\\_LE\\_CREATE\\_CONNECTION](#) command is outstanding

This command can be used at any time when address translation is disabled in the controller. When a controller cannot add a device to the resolving list because the list is full, it responds with error code 0x07 (memory capacity exceeded).

See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.38].

**Table 76. HCI\_LE\_ADD\_DEVICE\_TO\_RESOLVING\_LIST input parameters**

Parameter	Size	Description	Possible values
Peer_Identity_Address_Type	1	Identity address type	<ul style="list-style-type: none"><li>• 0x00: Public identity address</li><li>• 0x01: Random (static) identity address</li></ul>
Peer_Identity_Address	6	Public or random (static) identity address of the peer device	-
Peer_IRK	16	IRK of the peer device	-
Local_IRK	16	IRK of the local device	-

**Table 77. HCI\_LE\_ADD\_DEVICE\_TO\_RESOLVING\_LIST output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.1.47 HCI\_LE\_REMOVE\_DEVICE\_FROM\_RESOLVING\_LIST**

This command is used to remove a device from the list of address translations used to resolve private addresses in the controller. It cannot be used when address translation is enabled in the controller and:

- advertising is enabled
- scanning is enabled
- [HCI\\_LE\\_CREATE\\_CONNECTION](#) command is outstanding

This command can be used at any time when address translation is disabled in the controller. When a controller cannot remove a device from the resolving list because it is not found, it responds with error code 0x02 (unknown connection identifier).

See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.39].

**Table 78. HCI\_LE\_REMOVE\_DEVICE\_FROM\_RESOLVING\_LIST input parameters**

Parameter	Size	Description	Possible values
Peer_Identity_Address_Type	1	Identity address type	<ul style="list-style-type: none"><li>• 0x00: Public identity address</li><li>• 0x01: Random (static) identity address</li></ul>
Peer_Identity_Address	6	Public or random (static) identity address of the peer device	-

**Table 79. HCI\_LE\_REMOVE\_DEVICE\_FROM\_RESOLVING\_LIST output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.1.48 HCI\_LE\_CLEAR\_RESOLVING\_LIST**

This command is used to remove all devices from the list of address translations used to resolve resolvable private addresses in the controller. It cannot be used when address translation is enabled in the controller and:

- advertising is enabled
- scanning is enabled
- [HCI\\_LE\\_CREATE\\_CONNECTION](#) command is outstanding

This command can be used at any time when address translation is disabled in the controller. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.40].

**Input parameters:** none

**Table 80. HCI\_LE\_CLEAR\_RESOLVING\_LIST output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.1.49 HCI\_LE\_READ\_RESOLVING\_LIST\_SIZE**

This command is used to read the total number of address translation entries in the resolving list that can be stored in the controller. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.41].

**Input parameters:** none

**Table 81. HCI\_LE\_READ\_RESOLVING\_LIST\_SIZE output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-
Resolving_List_Size	1	Number of address translation entries in the resolving list	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

## 2.1.50 [HCI\\_LE\\_READ\\_PEER\\_RESOLVABLE\\_ADDRESS](#)

This command is used to get the current peer resolvable private address used for the corresponding peer public and random (static) identity address. The used peer resolvable address being can change after the command is called. This command can be used at any time. When a controller cannot find a resolvable private address associated with the peer identity address, it responds with error code 0x02 (unknown connection identifier). See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.42].

**Table 82. HCI\_LE\_READ\_PEER\_RESOLVABLE\_ADDRESS input parameters**

Parameter	Size	Description	Possible values
Peer_Identity_Address_Type	1	Identity address type	<ul style="list-style-type: none"><li>• 0x00: Public identity address</li><li>• 0x01: Random (static) identity address</li></ul>
Peer_Identity_Address	6	Public or random (static) identity address of the peer device	-

**Table 83. HCI\_LE\_READ\_PEER\_RESOLVABLE\_ADDRESS output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-
Peer_Resolvable_Address	6	Resolvable private address being used by the peer device	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

## 2.1.51 [HCI\\_LE\\_READ\\_LOCAL\\_RESOLVABLE\\_ADDRESS](#)

This command is used to get the current local resolvable private address used for the corresponding peer identity address. The used local resolvable address can change after the command is called. This command can be used at any time. When a controller cannot find a resolvable private address associated with the peer identity address, it responds with error code 0x02 (unknown connection identifier). See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.43].

**Table 84. HCI\_LE\_READ\_LOCAL\_RESOLVABLE\_ADDRESS input parameters**

Parameter	Size	Description	Possible values
Peer_Identity_Address_Type	1	Identity address type	<ul style="list-style-type: none"><li>• 0x00: Public identity address</li><li>• 0x01: Random (static) identity address</li></ul>
Peer_Identity_Address	6	Public or random (static) identity address of the peer device	-

**Table 85.** HCI\_LE\_READ\_LOCAL\_RESOLVABLE\_ADDRESS output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-
Local_Resolvable_Address	6	Resolvable private address being used by the local device	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.1.52****HCI\_LE\_SET\_ADDRESS\_RESOLUTION\_ENABLE**

This command is used to enable resolution of resolvable private addresses in the controller. This causes the controller to use the resolving list whenever it receives a local or peer resolvable private address. This command can be used at any time, except when:

- advertising is enabled
- acanning is enabled
- [HCI\\_LE\\_CREATE\\_CONNECTION](#) command is outstanding

See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.44].

**Table 86.** HCI\_LE\_SET\_ADDRESS\_RESOLUTION\_ENABLE input parameters

Parameter	Size	Description	Possible values
Address_Resolution_Enable	1	Enable/disable address resolution in the controller	<ul style="list-style-type: none"><li>• 0x00: address resolution disabled (default)</li><li>• 0x01: address resolution enabled</li></ul>

**Table 87.** HCI\_LE\_SET\_ADDRESS\_RESOLUTION\_ENABLE output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.1.53****HCI\_LE\_SET\_RESOLVABLE\_PRIVATE\_ADDRESS\_TIMEOUT**

This command sets the length of time the controller uses a resolvable private address, before a new resolvable private address is generated and starts to be used. This timeout applies to all addresses generated by the controller. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.45].

**Table 88.** HCI\_LE\_SET\_RESOLVABLE\_PRIVATE\_ADDRESS\_TIMEOUT input parameters

Parameter	Size	Description	Possible values
RPA_Timeout	2	RPA_Timeout, measured in seconds. Range 0x0001 to 0xA1B8 (1 s to ~11.5 hours), default 0x0384 (900 s)	-

**Table 89.** HCI\_LE\_SET\_RESOLVABLE\_PRIVATE\_ADDRESS\_TIMEOUT output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

## 2.1.54

**HCI\_LE\_READ\_MAXIMUM\_DATA\_LENGTH**

This command allows the host to read the controller maximum supported payload octets and packet duration times for transmission and reception (supportedMaxTxOctets, supportedMaxTxTime, supportedMaxRxOctets, and supportedMaxRxTime). See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.46].

**Input parameters:** none

**Table 90. HCI\_LE\_READ\_MAXIMUM\_DATA\_LENGTH output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-
supportedMaxTxOctets	2	Maximum number of payload octets that the local controller supports for transmission of a single link layer packet on a data connection	0x001B ... 0x00FB
supportedMaxTxTime	2	Maximum time, in microseconds, that the local controller supports for transmission of a single link layer packet on a data connection	0x0148 ... 0x4290
supportedMaxRxOctets	2	Maximum number of payload octets that the local controller supports for reception of a single link layer packet on a data connection	0x001B ... 0x00FB
supportedMaxRxTime	2	Maximum time, in microseconds, that the local controller supports for reception of a single link layer packet on a data connection	0x0148 ... 0x4290

**Events generated**

- **HCI\_COMMAND\_COMPLETE\_EVENT**

## 2.1.55

**HCI\_LE\_READ\_PHY**

This command is used to read the current transmitter and receiver PHY on the connection identified by the Connection\_Handle. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.47].

**Table 91. HCI\_LE\_READ\_PHY input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle for which the command is given	0x0000 ... 0x0EFF

**Table 92. HCI\_LE\_READ\_PHY output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-
Connection_Handle	2	Connection handle related to the response	0x0000 ... 0x0EFF
TX_PHY	1	Transmitter PHY in use	<ul style="list-style-type: none"><li>• 0x01: The transmitter PHY for the connection is LE 1M</li><li>• 0x02: The transmitter PHY for the connection is LE 2M</li><li>• 0x03: The transmitter PHY for the connection is LE coded (not supported by STM32WB)</li></ul>
RX_PHY	1	Receiver PHY in use	<ul style="list-style-type: none"><li>• 0x01: The receiver PHY for the connection is LE 1M</li><li>• 0x02: The receiver PHY for the connection is LE 2M</li><li>• 0x03: The receiver PHY for the connection is LE coded (not supported by STM32WB)</li></ul>

**Events generated**

- **HCI\_COMMAND\_COMPLETE\_EVENT**

## 2.1.56

**HCI\_LE\_SET\_DEFAULT\_PHY**

This command allows the host to specify its preferred values for the transmitter and receiver PHY, to be used for all subsequent connections over the LE transport.

The ALL\_PHYS parameter is a bit field that allows the host to specify, for each direction, whether it has no preference among the PHYs that the controller supports in a given direction, or whether there is a particular PHY that it prefers in the TX\_PHYS or RX\_PHYS parameter. The TX\_PHYS parameter is a bit field that indicates the transmitter PHYs that the host prefers the controller to use. If the ALL\_PHYS parameter specifies that the host has no preference, the TX\_PHYS parameter is ignored, otherwise at least one bit is set to 1. The RX\_PHYS parameter is a bit field that indicates the receiver PHYs that the host prefers the controller to use. If the ALL\_PHYS parameter specifies that the host has no preference, the RX\_PHYS parameter is ignored; otherwise at least one bit is set to 1.

See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.48].

**Table 93. HCI\_LE\_SET\_DEFAULT\_PHY input parameters**

Parameter	Size	Description	Possible values
ALL_PHYS	1	Host preferences for TX PHY and RX PHY	<ul style="list-style-type: none"> <li>• 0: the host has no preference among the transmitter PHYs supported by the controller</li> <li>• 1: the host has no preference among the receiver PHYs supported by the controller</li> </ul>
TX_PHYS	1	Host preferences for TX PHY	Bitmask of: <ul style="list-style-type: none"> <li>• 0x01: LE 1M PHY- preferred</li> <li>• 0x02: LE 2M PHY- preferred</li> <li>• 0x04: LE coded PHY-preferred (not supported by STM32WB)</li> </ul>
RX_PHYS	1	Host preferences for RX PHY	Bitmask of: <ul style="list-style-type: none"> <li>• 0x01: LE 1M PHY- preferred</li> <li>• 0x02: LE 2M PHY- preferred</li> <li>• 0x04: LE coded PHY-preferred (not supported by STM32WB)</li> </ul>

**Table 94. HCI\_LE\_SET\_DEFAULT\_PHY output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

#### Events generated

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

### 2.1.57 [HCI\\_LE\\_SET\\_PHY](#)

This command is used to set the PHY preferences for the connection identified by the Connection\_Handle. The controller can be unable to make the change (for example, when the peer does not support the requested PHY), or decide that the current PHY is the preferred one.

The ALL\_PHYS parameter is a bit field that allows the host to specify, for each direction, whether it has no preference among the PHYs that the controller supports in a given direction, or whether there is a PHY that it prefers in the TX\_PHYS or RX\_PHYS parameter. The TX\_PHYS/RX\_PHYS parameters are bit fields indicating to the transmitter/receiver PHYs that the Host prefers the controller to use. If the ALL\_PHYS parameter specifies that the Host has no preference, the TX\_PHYS/RX\_PHYS parameter is ignored, otherwise at least one bit is set to 1.

If, for at least one direction, the host has specified a preference and the current PHY is not among the preferred, the controller can request a change. The PHY preferences provided by the command override those provided via the HCI\_LE\_SET\_DEFAULT\_PHY command or any preferences previously set using this command on the same connection. The PHY\_options parameter is a bit field that allows the host to specify options for PHYs. The default value for a new connection is all 0. The controller can override any preferred coding for transmitting on the LE coded PHY. The host can specify a preferred coding, even if it prefers not to use the LE coded transmitter PHY, as the controller can override the PHY preference.

See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.49].

**Table 95. HCI\_LE\_SET\_PHY input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle for which the command is given	0x0000 ... 0x0EFF
ALL_PHYS	1	Host preferences for TX PHY and RX PHY	Bitmask of: • 0x01: LE 1M PHY- preferred • 0x02: LE 2M PHY- preferred • 0x04: LE Coded PHY-preferred (not supported by STM32WB)
TX_PHYS	1	Host preferences for TX PHY	Bitmask of: • 0x01: LE 1M PHY- preferred • 0x02: LE 2M PHY- preferred • 0x04: LE Coded PHY-preferred (not supported by STM32WB)
RX_PHYS	1	Host preferences for RX PHY	Bitmask of: • 0x01: LE 1M PHY- preferred • 0x02: LE 2M PHY- preferred • 0x04: LE Coded PHY-preferred (not supported by STM32WB)
PHY_options	2	Options for PHYs (not supported by STM32WB)	• 0x0000: the host has no preferred coding when transmitting on the LE coded PHY • 0x0001: the host prefers S = 2 coding when transmitting on the LE coded PHY • 0x0002: the host prefers S = 8 coding when transmitting on the LE coded PHY

**Table 96. HCI\_LE\_SET\_PHY output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_STATUS\\_EVENT](#)

**2.1.58**
**HCI\_LE\_SET\_ADVERTISING\_SET\_RANDOM\_ADDRESS**

This command is used by the Host to set the random device address specified by the Random\_Address parameter. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.52].

**Table 97. HCI\_LE\_SET\_ADVERTISING\_SET\_RANDOM\_ADDRESS input parameters**

Parameter	Size	Description	Possible values
Advertising_Handle	1	Used to identify an advertising set	0x00 ... 0xEF
Random_Address	6	Random device address	-

**Table 98. HCI\_LE\_SET\_ADVERTISING\_SET\_RANDOM\_ADDRESS output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

## 2.1.59

**HCI\_LE\_SET\_EXTENDED\_ADVERTISING\_PARAMETERS**

This command is used by the Host to set the extended advertising parameters. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.53].

**Table 99. HCI\_LE\_SET\_EXTENDED\_ADVERTISING\_PARAMETERS input parameters**

Parameter	Size	Description	Possible values
Advertising_Handle	1	Used to identify an advertising set	0x00 ... 0xEF
Adv_Event_Properties	2	Type of advertising event.	Bitmask of: <ul style="list-style-type: none"><li>• 0x0001: Connectable advertising</li><li>• 0x0002: Scannable advertising</li><li>• 0x0004: Directed advertising</li><li>• 0x0008: High duty-cycle directed connectable advertising</li><li>• 0x0010: Use legacy advertising PDUs</li><li>• 0x0020: Anonymous advertising</li><li>• 0x0040: Include Tx Power in at least one advertising PDU</li></ul>
Primary_Adv_Interval_Min	3	Minimum advertising interval. Time = N * 0.625 ms	0x000020 (20.000 ms) ... 0xFFFFFFF (10485759.375 ms)
Primary_Adv_Interval_Max	3	Maximum advertising interval. Time = N * 0.625 ms	0x000020 (20.000 ms) ... 0xFFFFFFF (10485759.375 ms)
Primary_Adv_Channel_Map	1	Advertising channel map	Bitmask of: <ul style="list-style-type: none"><li>• 0x01: Use Channel 37</li><li>• 0x02: Use Channel 38</li><li>• 0x04: Use Channel 39</li></ul>
Own_Address_Type	1	Own address type	<ul style="list-style-type: none"><li>• 0x00: Public device address</li><li>• 0x01: Random device address</li><li>• 0x02: Resolvable private address if available, otherwise Public address</li><li>• 0x03: Resolvable private address if available, otherwise Random address</li></ul>
Peer_Address_Type	1	Address type of the peer device	<ul style="list-style-type: none"><li>• 0x00: Public device address or Public identity address</li><li>• 0x01: Random device address or Random (static) identity address</li></ul>
Peer_Address	6	Public device address, Random device address, Public identity address, or Random (static) identity address of the device to be connected.	-
Adv_Filter_Policy	1	Advertising filter policy	<ul style="list-style-type: none"><li>• 0x00: Process scan and connection requests from all devices (White list not in use)</li><li>• 0x01: Process connection requests from all devices and scan requests only from devices in the White list.</li><li>• Process scan requests from all devices and connection requests only from devices in the White list.</li><li>• 0x03: Process scan and connection requests only from devices in the White List.</li></ul>
Adv_TX_Power	1	Advertising TX power. Units: dBm.	<ul style="list-style-type: none"><li>• -127 ... 20</li><li>• 127: Host has no preferences</li></ul>

Parameter	Size	Description	Possible values
Primary_Adv_PHY	1	Primary advertising PHY	<ul style="list-style-type: none"> <li>• 0x01: Primary advertisement PHY is LE 1M</li> <li>• 0x03: Primary advertisement PHY is LE coded (not supported by STM32WB devices)</li> </ul>
Secondary_Adv_Max_Skip	1	Secondary advertising maximum skip	<ul style="list-style-type: none"> <li>• 0x00: AUX_ADV_IND must be sent prior to the next advertising event</li> <li>• 0x01 ... 0xFF: Maximum advertising events that the controller can skip before sending the AUX_ADV_IND packets on the secondary advertising physical channel</li> </ul>
Secondary_Adv_PHY	1	Secondary advertising PHY	<ul style="list-style-type: none"> <li>• 0x01: Secondary advertisement PHY is LE 1M</li> <li>• 0x02: Secondary advertisement PHY is LE 2M</li> <li>• 0x03: Secondary advertisement PHY is LE coded (not supported by STM32WB devices)</li> </ul>
Adv_SID	1	Value of the Advertising SID subfield in the ADLfield of the PDU	<ul style="list-style-type: none"> <li>• 0x00 ... 0x0F</li> </ul>
Scan_Req_Notification_Enable	1	Scan request notifications	<ul style="list-style-type: none"> <li>• 0x00: Scan request notifications disabled</li> <li>• 0x01: Scan request notifications enabled</li> </ul>

**Table 100. HCI\_LE\_SET\_EXTENDED\_ADVERTISING\_PARAMETERS output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-
Selected_TX_Power	1	Power level selected by the controller. Units: dBm.	<ul style="list-style-type: none"> <li>• -127 ... 20</li> </ul>

#### Events generated

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

#### 2.1.60 [HCI\\_LE\\_SET\\_EXTENDED\\_ADVERTISING\\_DATA](#)

This command is used to set the data used in extended advertising PDUs with a data field. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.54].

**Table 101. HCI\_LE\_SET\_EXTENDED\_ADVERTISING\_DATA input parameters**

Parameter	Size	Description	Possible values
Advertising_Handle	1	Used to identify an advertising set	0x00 ... 0xEF
Operation	1	Advertising operation	<ul style="list-style-type: none"> <li>• 0x00: intermediate fragment of fragmented extended advertising data</li> <li>• 0x01: first fragment of fragmented extended advertising data</li> <li>• 0x02: last fragment of fragmented extended advertising data</li> <li>• 0x03: complete extended advertising data</li> <li>• 0x04: unchanged data (just update the advertising DID)</li> </ul>
Fragment_Preference	1	Fragment preference	<ul style="list-style-type: none"> <li>• 0x00: the controller can fragment all data</li> <li>• 0x02: the controller should not fragment data, or minimize fragmentation</li> </ul>

Parameter	Size	Description	Possible values
Advertising_Data_Length	1	Length of Advertising_Data in octets	-
Advertising_Data	Advertising_Data_Length	Data formatted as defined in Bluetooth spec. v.5.2 [Vol 3, Part C, 11].	-

**Table 102.** HCI\_LE\_SET\_EXTENDED\_ADVERTISING\_DATA output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.1.61****HCI\_LE\_SET\_EXTENDED\_SCAN\_RESPONSE\_DATA**

This command is used by to provide scan response data used in scanning response PDUs during extended advertising. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.55].

**Table 103.** HCI\_LE\_SET\_EXTENDED\_SCAN\_RESPONSE\_DATA input parameters

Parameter	Size	Description	Possible values
Advertising_Handle	1	Used to identify an advertising set	0x00 ... 0xEF
Operation	1	Scan response operation	<ul style="list-style-type: none"> <li>• 0x00: intermediate fragment of fragmented scan response data</li> <li>• 0x01: first fragment of fragmented scan response data</li> <li>• 0x02: last fragment of fragmented scan response data</li> <li>• 0x03: complete scan response data</li> </ul>
Fragment_Preference	1	Fragment preference	<ul style="list-style-type: none"> <li>• 0x00: the controller can fragment all data</li> <li>• 0x02: the controller should not fragment data, or minimize fragmentation</li> </ul>
Scan_Response_Data_Length	1	Length of Scan_Response_Data in octets	-
Scan_Response_Data	Scan_Response_Data_Length	Data formatted as defined in Bluetooth spec. v.5.2 [Vol 3, Part C, 11].	-

**Table 104.** HCI\_LE\_SET\_EXTENDED\_SCAN\_RESPONSE\_DATA output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

## 2.1.62

**HCI\_LE\_SET\_EXTENDED\_ADVERTISING\_ENABLE**

This command is used to request the controller to enable/disable one or more advertising sets using those identified by the Advertising\_Handle[i] parameter. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.56].

**Table 105. HCI\_LE\_SET\_EXTENDED\_ADVERTISING\_ENABLE input parameters**

Parameter	Size	Description	Possible values
Enable	1	Enable/disable advertising	<ul style="list-style-type: none"><li>• 0x00: advertising disabled</li><li>• 0x01: advertising enabled</li></ul>
Num_Sets	1	Number of advertising sets	<ul style="list-style-type: none"><li>• 0x00: disable all advertising sets</li><li>• 0x00 ... 0x3F: number of advertising sets to enable/disable</li></ul>
Advertising_Handle[i]	1	Used to identify an advertising set	0x00 ... 0xEF
Duration[i]	2	Duration of advertising set. Time = N * 10 ms.	<ul style="list-style-type: none"><li>• 0x0000 (0 ms) : No advertising duration</li><li>• 0x0001 (10 ms) ... 0xFFFF (655350 ms): advertising duration</li></ul>
Max_Extended_Advertising_Events[i]	1	Maximum number of advertising events	<ul style="list-style-type: none"><li>• 0x00 (0 ms) : No maximum number</li><li>• 0x01 ... 0xFF: maximum number of events the controller must try to send before terminating the extended advertising</li></ul>

**Table 106. HCI\_LE\_SET\_EXTENDED\_ADVERTISING\_ENABLE output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)
- [HCI\\_LE\\_ENHANCED\\_CONNECTION\\_COMPLETE\\_EVENT](#)

## 2.1.63

**HCI\_LE\_READ\_MAXIMUM\_ADVERTISING\_DATA\_LENGTH**

This command is used by the Host to set the random device address specified by the Random\_Address parameter. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.57].

**Input parameters:** none

**Table 107. HCI\_LE\_READ\_MAXIMUM\_ADVERTISING\_DATA\_LENGTH output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-
Max_Advertising_Data_Length	2	Maximum supported advertising data length	0x001F ... 0x0672

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

## 2.1.64

**HCI\_LE\_READ\_NUMBER\_OF\_SUPPORTED\_ADVERTISING\_SETS**

This command is used by the Host to set the random device address specified by the Random\_Address parameter. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.58].

**Input parameters:** none

**Table 108. HCI\_LE\_READ\_NUMBER\_OF\_SUPPORTED\_ADVERTISING\_SETS output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-
Num_Supported_Advertising_Sets	1	Number of advertising sets supported at the same time	0x01 ... 0xF0

**Events generated**

- `HCI_COMMAND_COMPLETE_EVENT`

**2.1.65****HCI\_LE\_REMOVE\_ADVERTISING\_SET**

This command is used to remove an advertising set from the controller. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.59].

**Table 109. HCI\_LE\_REMOVE\_ADVERTISING\_SET input parameters**

Parameter	Size	Description	Possible values
Advertising_Handle	1	Used to identify an advertising set	0x00 ... 0xEF

**Table 110. HCI\_LE\_REMOVE\_ADVERTISING\_SET output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- `HCI_COMMAND_COMPLETE_EVENT`

**2.1.66****HCI\_LE\_CLEAR\_ADVERTISING\_SETS**

This command is used to remove all existing advertising sets from the controller. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.60].

**Input parameters:** none

**Table 111. HCI\_LE\_CLEAR\_ADVERTISING\_SETS output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- `HCI_COMMAND_COMPLETE_EVENT`

**2.1.67****HCI\_LE\_SET\_EXTENDED\_SCAN\_PARAMETERS**

This command is used to set the extended scan parameters to be used on the advertising physical channels. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.64].

**Table 112. HCI\_LE\_SET\_EXTENDED\_SCAN\_PARAMETERS input parameters**

Parameter	Size	Description	Possible values
Own_Address_Type	1	Own address type	<ul style="list-style-type: none"><li>• 0x00: public device address</li><li>• 0x01: random device address</li><li>• 0x02: resolvable private address if available, otherwise public address</li><li>• 0x03: resolvable private address if available, otherwise random address</li></ul>

Parameter	Size	Description	Possible values
Scanning_Filter_Policy	1	Scan filter policy	<ul style="list-style-type: none"> <li>• 0x00: accept all advertising and scan response PDUs, except directed advertising PDUs not addressed to this device.</li> <li>• 0x01: accept only advertising and scan response PDUs from devices where the advertiser's address is in the White list. Directed advertising PDUs not addressed to this device are ignored.</li> <li>• 0x02: accept all advertising and scan response PDUs, except directed advertising PDUs where the identity address corresponding to TargetA does not address this device.</li> <li>• 0x03: accept all advertising and scan response PDUs, except advertising and scan response PDUs where the advertiser's identity address is not in the White list, and directed advertising PDUs where the identity address corresponding to TargetA does not address this device.</li> </ul>
Scanning_PHYs	1	Scan PHYs	Bitmask of: <ul style="list-style-type: none"> <li>• 0x01: scan advertisements on the LE 1M PHY</li> <li>• 0x04: scan advertisements on the LE Coded PHY (not supported by STM32WB devices)</li> </ul>
Scan_Type	1	Passive or active scanning. With passive scanning no scan request PDUs are sent.	<ul style="list-style-type: none"> <li>• 0x00: passive scanning</li> <li>• 0x01: active scanning</li> </ul>
Scan_Interval	2	Time interval from when the controller started its last scan until it begins the subsequent scan on the primary advertising physical channel. Time = N * 0.625 ms.	0x0004 (2.5 ms) ... 0x5DC0 (15000.0 ms)
Scan_Window	2	Duration of the scan on the primary advertising physical channel. Time = N * 0.625 ms.	0x0004 (2.5 ms) ... 0x5DC0 (15000.0 ms)

**Table 113. HCI\_LE\_SET\_EXTENDED\_SCAN\_PARAMETERS output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

#### Events generated

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

#### 2.1.68 [HCI\\_LE\\_SET\\_EXTENDED\\_SCAN\\_ENABLE](#)

This command is used to enable/disable extended scanning. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.65].

**Table 114. HCI\_LE\_SET\_EXTENDED\_SCAN\_ENABLE input parameters**

Parameter	Size	Description	Possible values
Enable	1	Enable/disable scan	<ul style="list-style-type: none"> <li>• 0x00: scanning disabled</li> <li>• 0x01: scanning enabled</li> </ul>
Filter_Duplicates	1	Duplicate filtering	<ul style="list-style-type: none"> <li>• 0x00: duplicate filtering disabled</li> <li>• 0x01: duplicate filtering enabled</li> <li>• 0x02: duplicate filtering enabled, reset for each scan period</li> </ul>
Duration	2	Scan duration. Time = N * 10 ms.	<ul style="list-style-type: none"> <li>• 0x0000 (0 ms): scan continuously until explicitly disabled</li> </ul>

Parameter	Size	Description	Possible values
			<ul style="list-style-type: none"> <li>• 0x0001 (10 ms) ... 0xFFFF (655350 ms)</li> </ul>
Period	2	Scan period. Time = N * 1.28 s.	<ul style="list-style-type: none"> <li>• 0x0000 (0 ms): scan continuously</li> <li>• 0x0001 (1280 ms) ... 0xFFFF (83884800 ms): time interval from when the controller started its last Scan_Duration until it begins the subsequent one</li> </ul>

**Table 115. HCI\_LE\_SET\_EXTENDED\_SCAN\_ENABLE output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)
- [HCI\\_LE\\_EXTENDED\\_ADVERTISING\\_REPORT\\_EVENT](#)

**2.1.69****HCI\_LE\_EXTENDED\_CREATE\_CONNECTION**

This command is used to create an ACL connection to a connectable advertiser by means of extended scanning. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.66].

**Table 116. HCI\_LE\_EXTENDED\_CREATE\_CONNECTION input parameters**

Parameter	Size	Description	Possible values
Initiator_Filter_Policy	1	Initiator filter policy	<ul style="list-style-type: none"> <li>• 0x00: White list not used</li> <li>• 0x01: White list used</li> </ul>
Own_Address_Type	1	Own address type	<ul style="list-style-type: none"> <li>• 0x00: public device</li> <li>• 0x01: random device</li> <li>• 0x02: resolvable private address if available, otherwise public address</li> <li>• 0x03: resolvable public address if available, otherwise random address</li> </ul>
Peer_Address_Type	1	Address type of the peer device	<ul style="list-style-type: none"> <li>• 0x00: public device or public identity address</li> <li>• 0x01: random device or random (static) identity address</li> </ul>
Peer_Address	6	Public or random device address, public or random (static) identity address of the device to connect	-
Initiating_PHYs	1	Initiating PHYs	Bitmask of: <ul style="list-style-type: none"> <li>• 0x01: scan connectable advertisements on the LE 1M PHY - Connection parameters for the LE 1M PHY</li> <li>• 0x02: connection parameters for the LE 2M PHY</li> <li>• 0x04: scan connectable advertisements on the LE coded PHY (not supported on STM32WB)</li> </ul>
Scan_Interval	2	Time interval from when the controller started its last scan until it begins the subsequent scan on the primary advertising physical channel. Time = N * 0.625 ms.	0x0004 (2.5 ms) ... 0x5DC0 (15000.0 ms)
Scan_Window	2	Duration of the scan on the primary advertising physical channel. Time = N * 0.625 ms.	0x0004 (2.5 ms) ... 0x5DC0 (15000.0 ms)

Parameter	Size	Description	Possible values
Conn_Interval_Min	2	Minimum value of the connection event interval (must be lower than or equal to Conn_Interval_Max. Time = N * 1.25 ms).	0x0006 (7.5 ms) ... 0x0C80 (4000.0 ms)
Conn_Interval_Max	2	Maximum value of the connection event interval (must be higher than or equal to Conn_Interval_Min. Time = N * 1.25 ms).	0x0006 (7.5 ms) ... 0x0C80 (4000.0 ms)
Conn_Latency	2	Peripheral latency for the connection, in number of connection events.	0x0000 ... 0x01F3
Supervision_Timeout	2	Supervision timeout for the LE Link. Must be a multiple of 10 ms and larger than $(1 + \text{connPeripheralLatency}) * \text{connInterval} * 2$ . Time = N * 10 ms.	0x000A (100 ms) ... 0x0C80 (32000 ms)
Min_CE_Length	2	Minimum length needed for this LE connection. Time = N * 0.625 ms.	0x0000 (0.0 ms) ... 0xFFFF (40959.375 ms)
Max_CE_Length	2	Maximum length needed for this LE connection. Time = N * 0.625 ms.	0x0000 (0.0 ms) ... 0xFFFF (40959.375 ms)

**Table 117. HCI\_LE\_EXTENDED\_CREATE\_CONNECTION output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

#### Events generated

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)
- [HCI\\_LE\\_CONNECTION\\_COMPLETE\\_EVENT](#)

### 2.1.70 [HCI\\_LE\\_READ\\_TRANSMIT\\_POWER](#)

This command is used to read the minimum and maximum transmit power supported by the controller. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.74].

**Input parameters** : none

**Table 118. HCI\_LE\_READ\_TRANSMIT\_POWER output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-
Min_TX_Power	1	Signed integer, in dBm	-127 ... 20
Max_TX_Power	1	Signed integer, in dBm	-127 ... 20

#### Events generated

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

### 2.1.71 [HCI\\_LE\\_READ\\_RF\\_PATH\\_COMPENSATION](#)

This command is used to read the RF path compensation value parameters used in the Tx power level and RSSI calculation. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.75].

**Input parameters**: none

**Table 119. HCI\_LE\_READ\_RF\_PATH\_COMPENSATION output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-
RF_TX_Path_Compensation	2	RF TX path compensation value (16-bit signed integer), in 0.1 dB units	-1280 ... 1280
RF_RX_Path_Compensation	2	RF RX path compensation value (16-bit signed integer), in 0.1 dB units	-1280 ... 1280

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.1.72**
**HCI\_LE\_WRITE\_RF\_PATH\_COMPENSATION**

This command is used to indicate the contribution of intermediate components in the RF path to the gain or loss between the RF transceiver and the antenna. Positive/negative values mean a net RF path gain/loss. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.76].

**Table 120. HCI\_LE\_WRITE\_RF\_PATH\_COMPENSATION input parameters**

Parameter	Size	Description	Possible values
RF_TX_Path_Compensation	2	RF TX path compensation value (16-bit signed integer), in 0.1 dB units	-1280 ... 1280
RF_RX_Path_Compensation	2	RF RX path compensation value (16-bit signed integer), in 0.1 dB units	-1280 ... 1280

**Table 121. HCI\_LE\_WRITE\_RF\_PATH\_COMPENSATION output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.1.73**
**HCI\_LE\_SET\_PRIVACY\_MODE**

This command is used to allow the host to specify the privacy mode to use for a given entry on the resolving list. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.74].

**Table 122. HCI\_LE\_SET\_PRIVACY\_MODE input parameters**

Parameter	Size	Description	Possible values
Peer_Identity_Address_Type	1	Identity address type	<ul style="list-style-type: none"> <li>• 0x00: Public identity address</li> <li>• 0x01: Random (static) identity address</li> </ul>
Peer_Identity_Address	6	Public or random (static) identity address of the peer device	-
Privacy Mode	1	Privacy mode	<ul style="list-style-type: none"> <li>• 0x00: Use network privacy mode</li> <li>• 0x01: Use device privacy mode</li> </ul>

**Table 123. HCI\_LE\_SET\_PRIVACY\_MODE output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- `HCI_COMMAND_COMPLETE_EVENT`

**2.1.74****`HCI_LE_GENERATE_DHKEY_V2`**

This command is used to start the generation of a Diffie-Hellman key in the controller for use over the LE transport. It takes the remote P-256 public key as input, and uses the private key generated by the `HCI_LE_READ_LOCAL_P256_PUBLIC_KEY` command or the private debug key. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.37].

**Table 124. `HCI_LE_GENERATE_DHKEY_V2` input parameters**

Parameter	Size	Description	Possible values
Remote_P256_Public_Key	64	The remote P-256 public key in X, Y format: <ul style="list-style-type: none"><li>• Octets 31-0: X coordinate</li><li>• Octets 63-32: Y coordinate, little Endian format</li></ul>	-
Key_Type	1	Type of private key used for the Diffie-Hellman key generation	<ul style="list-style-type: none"><li>• 0x00: Use the generated private key</li><li>• 0x01: Use the debug private key</li></ul>

**Table 125. `HCI_LE_GENERATE_DHKEY_V2` output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- `HCI_COMMAND_COMPLETE_EVENT`
- `HCI_LE_GENERATE_DHKEY_COMPLETE_EVENT`

## 2.2

## HCI testing commands

In Table 126 "Y" means that the corresponding command applies to the dedicated BLE stack variant, namely LO / BO / PO / BF/ LB, respectively, for Link layer only / Beacon only / Peripheral only / Basic feature / Link layer only basic.

Table 126. HCI testing commands list

Command	OpCode	LO	PO	BO	BF	LB
HCI_LE_RECEIVER_TEST	0x201D	Y	Y	Y	Y	Y
HCI_LE_TRANSMITTER_TEST	0x201E	Y	Y	Y	Y	Y
HCI_LE_TEST_END	0x201F	Y	Y	Y	Y	Y
HCI_LE_RECEIVER_TEST_V2	0x2033	Y	-	-	Y	Y
HCI_LE_TRANSMITTER_TEST_V2	0x2034	Y	-	-	Y	Y

### 2.2.1

#### HCI\_LE\_RECEIVER\_TEST

This command is used to start a test where the DUT receives test reference packets at a fixed interval from the tester. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.28].

Table 127. HCI\_LE\_RECEIVER\_TEST input parameters

Parameter	Size	Description	Possible values
RX_Frequency	1	N = (F - 2402) / 2 - Frequency range: 2402 to 2480 MHz	0x00 ... 0x27

Table 128. HCI\_LE\_RECEIVER\_TEST output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-

#### Events generated

- HCI\_COMMAND\_COMPLETE\_EVENT

### 2.2.2

#### HCI\_LE\_TRANSMITTER\_TEST

This command is used to start a test where the DUT generates test reference packets at a fixed interval. The controller transmits at maximum power. An LE controller supporting the command supports Packet\_Payload values 0x00, 0x01, and 0x02. An LE controller supports other values. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.29].

Table 129. HCI\_LE\_TRANSMITTER\_TEST input parameters

Parameter	Size	Description	Possible values
TX_Frequency	1	N = (F - 2402) / 2 - Frequency range: 2402 to 2480 MHz	0x00 ... 0x27
Length_Of_Test_Data	1	Length in bytes of payload data in each packet.	<ul style="list-style-type: none"><li>0x00 ... 0x25: BO variant</li><li>0x00 ... 0xFF: otherwise</li></ul>
Packet_Payload	1	Type of packet payload.	<ul style="list-style-type: none"><li>0x00: Pseudo-random bit sequence 9</li><li>0x01: Pattern of alternating bits 11110000</li><li>0x02: Pattern of alternating bits 10101010</li><li>0x03: Pseudo-random bit sequence 15</li><li>0x04: Pattern of all 1 bits</li><li>0x05: Pattern of all 0 bits</li><li>0x06: Pattern of alternating bits 00001111</li><li>0x07: Pattern of alternating bits 0101</li></ul>

**Table 130. HCI\_LE\_TRANSMITTER\_TEST output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_STATUS\\_EVENT](#)
- [HCI\\_DISCONNECTION\\_COMPLETE\\_EVENT](#)

**2.2.3**
**HCI\_LE\_TEST\_END**

This command is used to stop any test in progress. The Number\_Of\_Packets is an unsigned number and contains the number of received packets, for a transmitter test is 0x0000. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.30].

**Input parameters:** none

**Table 131. HCI\_LE\_TEST\_END output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-
Number_Of_Packets	2	Number of packets received	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.2.4**
**HCI\_LE\_RECEIVER\_TEST\_V2**

This command is used to start a test where the DUT receives test reference packets at a fixed interval. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.28].

**Table 132. HCI\_LE\_RECEIVER\_TEST\_V2 input parameters**

Parameter	Size	Description	Possible values
RX_Frequency	1	N = (F - 2402) / 2 - Frequency range: 2402 to 2480 MHz	0x00 ... 0x27
PHY	1	PHY to use for test packet	<ul style="list-style-type: none"> <li>• 0x00: Reserved for future use</li> <li>• 0x01: Transmitter set to use the LE 1M PHY</li> <li>• 0x02: Transmitter set to use the LE 2M PHY</li> <li>• 0x03: Transmitter set to use the LE coded PHY with S = 8 data coding (not supported by STM32WB devices)</li> <li>• 0x04: Transmitter set to use the LE coded PHY with S = 2 data coding (not supported by STM32WB devices)</li> </ul>
Modulation_Index	1	Modulation index capability of the transmitter	<ul style="list-style-type: none"> <li>• 0x00: Assume transmitter has a standard modulation index</li> <li>• 0x01: Assume transmitter has a stable modulation index</li> </ul>

**Table 133. HCI\_LE\_RECEIVER\_TEST\_V2 output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

## 2.2.5 HCI\_LE\_TRANSMITTER\_TEST\_V2

This command is used to start a test where the DUT generates test reference packets at a fixed interval. The controller transmits at maximum power. An LE controller supporting this command supports Packet\_Payload values 0x00, 0x01 and 0x02. An LE controller supporting the LE coded PHY also supports Packet\_Payload value 0x04 (not supported by STM32WB MCUs). An LE controller may support other values of Packet\_Payload. See Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.29].

**Table 134. HCI\_LE\_TRANSMITTER\_TEST\_V2 input parameters**

Parameter	Size	Description	Possible values
TX_Frequency	1	N = (F - 2402) / 2 - Frequency range: 2402 to 2480 MHz	0x00 ... 0x27
Length_Of_Test_Data	1	Length in bytes of payload data in each packet.	0x00 ... 0x25
Packet_Payload	1	Type of packet payload.	<ul style="list-style-type: none"><li>• 0x00: Pseudo-random bit sequence 9</li><li>• 0x01: Pattern of alternating bits 11110000</li><li>• 0x02: Pattern of alternating bits 10101010</li><li>• 0x03: Pseudo-random bit sequence 15</li><li>• 0x04: Pattern of all 1 bits</li><li>• 0x05: Pattern of all 0 bits</li><li>• 0x06: Pattern of alternating bits 00001111</li><li>• 0x07: Pattern of alternating bits 0101</li></ul>
PHY	1	PHY to use for test packet	<ul style="list-style-type: none"><li>• 0x00: Reserved for future use</li><li>• 0x01: Transmitter set to use the LE 1M PHY</li><li>• 0x02: Transmitter set to use the LE 2M PHY</li><li>• 0x03: Transmitter set to use the LE coded PHY with S = 8 data coding</li><li>• 0x04: Transmitter set to use the LE coded PHY with S = 2 data coding</li></ul>

**Table 135. HCI\_LE\_TRANSMITTER\_TEST\_V2 output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

### Events generated

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

## 2.3 HAL commands

In Table 136 "Y" means that the corresponding command applies to the dedicated BLE stack variant, namely LO / BO / PO / BF / LB, respectively, for Link layer only / Beacon only / Peripheral only / Basic feature / Link layer only basic.

Table 136. HAL commands list

Command	OpCode	LO	PO	BO	BF	LB
ACI_HAL_GET_FW_BUILD_NUMBER	0xFC00	Y	Y	Y	Y	Y
ACI_HAL_WRITE_CONFIG_DATA	0xFC0C	Y	Y	Y	Y	Y
ACI_HAL_READ_CONFIG_DATA	0xFC0D	Y	Y	Y	Y	Y
ACI_HAL_SET_TX_POWER_LEVEL	0xFC0F	Y	Y	Y	Y	Y
ACI_HAL_LE_TX_TEST_PACKET_NUMBER	0xFC14	Y	Y	-	Y	Y
ACI_HAL_TONE_START	0xFC15	Y	Y	Y	Y	Y
ACI_HAL_TONE_STOP	0xFC16	Y	Y	Y	Y	Y
ACI_HAL_GET_LINK_STATUS	0xFC17	Y	Y	-	Y	Y
ACI_HAL_SET_RADIO_ACTIVITY_MASK	0xFC18	Y	Y	Y	Y	Y
ACI_HAL_GET_ANCHOR_PERIOD	0xFC19	Y	Y	Y	Y	Y
ACI_HAL_SET_EVENT_MASK	0xFC1A	-	-	-	-	-
ACI_HAL_GET_PM_DEBUG_INFO	0xFC1C	-	-	-	-	-
ACI_HAL_SET_PERIPHERAL_LATENCY	0xFC20	Y	-	-	Y	Y
ACI_HAL_READ_RSSI	0xFC22	Y	Y	Y	Y	Y
ACI_HAL_READ_RADIO_REG	0xFC30	-	-	-	-	-
ACI_HAL_WRITE_RADIO_REG	0xFC31	-	-	-	-	-
ACI_HAL_READ_RAW_RSSI	0xFC32	-	-	-	-	-
ACI_HAL_RX_START	0xFC33	-	-	-	-	-
ACI_HAL_RX_STOP	0xFC34	-	-	-	-	-
ACI_HAL_STACK_RESET	0xFC3B	Y	Y	Y	Y	Y

### 2.3.1 ACI\_HAL\_GET\_FW\_BUILD\_NUMBER

This command returns the build number associated with the firmware version currently running.

**Input parameters:** none

Table 137. ACI\_HAL\_GET\_FW\_BUILD\_NUMBER output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-
Build_Number	2	Build number of the firmware.	-

#### Events generated

- HCI\_COMMAND\_COMPLETE\_EVENT

### 2.3.2 ACI\_HAL\_WRITE\_CONFIG\_DATA

This command writes a value to a low level configure data structure. It is useful to set up directly some low level parameters for the system in the runtime.

Note: The static random address set by this command is taken into account by the GAP only when it receives the ACI\_GAP\_INIT command.

**Table 138. ACI\_HAL\_WRITE\_CONFIG\_DATA input parameters**

Parameter	Size	Description	Possible values
Offset	1	Offset of the element to write in the configuration data structure	<ul style="list-style-type: none"> <li>0x00: CONFIG_DATA_PUBADDR_OFFSET, Bluetooth public address, 6 bytes</li> <li>0x08: CONFIG_DATA_ER_OFFSET, encryption root key used to derive LTK and CSRK; 16 bytes</li> <li>0x18: CONFIG_DATA_IR_OFFSET, identity root key used to derive LTK and CSRK, 16 bytes</li> <li>0x2E: CONFIG_DATA_RANDOM_ADDRESS_WR, static random address, 6 bytes</li> <li>0x34: CONFIG_DATA_GAP_ADD_REC_NBR_OFFSET; GAP service additional record number</li> <li>0x35: CONFIG_DATA_SC_KEY_TYPE_OFFSET; secure connections key type (0: normal, 1: debug), 1 byte</li> <li>0xB0: CONFIG_DATA_SMP_MODE_OFFSET; SMP mode (0: normal, 1: bypass, 2: no blacklist), 1 byte</li> <li>0xC0: CONFIG_DATA_LL_SCAN_CHAN_MAP_OFFSET, LL scan channel map (same format as Primary_Adv_Channel_Map), 1 byte</li> <li>0xC1: CONFIG_DATA_LL_BG_SCAN_MODE_OFFSET, LL background scan mode (0: BG scan disabled, 1: BG scan enabled), 1 byte</li> </ul>
Length	1	Length of data to be written	-
Value	Length	Data to be written	-

**Table 139. ACI\_HAL\_WRITE\_CONFIG\_DATA output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

#### Events generated

- HCI\_COMMAND\_COMPLETE\_EVENT

#### 2.3.3 ACI\_HAL\_READ\_CONFIG\_DATA

This command requests the value in the low level configure data structure. The number of read bytes changes for different offsets.

**Table 140. ACI\_HAL\_READ\_CONFIG\_DATA input parameters**

Parameter	Size	Description	Possible values
Offset	1	<p>Offset of the element in the configuration data structure to be read. The valid offsets are:</p> <ul style="list-style-type: none"> <li>0x00: Bluetooth public address, returned value length is 6 bytes</li> <li>0x08: Encryption root key used to derive LTK and CSRK, returned value length is 16 bytes</li> <li>0x18: Identity root key used to derive LTK and CSRK, returned value length is 16 bytes</li> <li>0x80: Static random address, returned value length is 6 bytes (read-only)</li> </ul>	<ul style="list-style-type: none"> <li>0x00: CONFIG_DATA_PUBADDR_OFFSET</li> <li>0x08: CONFIG_DATA_ER_OFFSET</li> <li>0x18: CONFIG_DATA_IR_OFFSET</li> <li>0x80: CONFIG_DATA_RANDOM_ADDRESS</li> </ul>

**Table 141.** ACI\_HAL\_READ\_CONFIG\_DATA output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-
Data_Length	1	Length of Data in octets	-
Data	Data_Length	Data field associated with Offset parameter	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.3.4****ACI\_HAL\_SET\_TX\_POWER\_LEVEL**

This command sets the Tx power level of the device by controlling the PA\_LEVEL that determines the output power level (dBm) at the IC pin.

When the system starts up or reboots, the default Tx power level (the maximum value of 6 dBm) is used. Once this command is given, the output power is changed instantaneously, regardless if there is an ongoing Bluetooth communication or not. For example, for debugging purpose, the device can be set to advertise all the time, and use this command to observe the change of signal strength. The system keeps the last received power level from the command, i.e. the second command overwrites the previous power level. The new power level remains until the next ACI\_HAL\_SET\_TX\_POWER\_LEVEL command, or the system reboots.

The advertising extensions commands allow, per advertising set, to override the value determined by the ACI\_HAL\_SET\_TX\_POWER\_LEVEL command (see [ACI\\_GAP\\_ADV\\_SET\\_CONFIGURATION](#)). Refer to [Section 5 Tx power level](#) for the values of PA\_Level parameter.

**Table 142.** ACI\_HAL\_SET\_TX\_POWER\_LEVEL input parameters

Parameter	Size	Description	Possible values
En_High_Power	1	Enable high power mode (deprecated and ignored on STM32WB)	<ul style="list-style-type: none"><li>• 0x00: Standard power</li><li>• 0x01: High power</li></ul>
PA_Level	1	Power amplifier output level.	0x00 ... 0x23

**Table 143.** ACI\_HAL\_SET\_TX\_POWER\_LEVEL output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.3.5****ACI\_HAL\_LE\_TX\_TEST\_PACKET\_NUMBER**

This command returns the number of packets sent in direct test mode. When the direct TX test is started, a 16-bit counter is used to count the transmitted packets, starting from 0, and counting upwards. The counter can wrap and start from 0 again. The counter is not cleared until the next direct TX test starts.

**Input parameters:** none

**Table 144.** ACI\_HAL\_LE\_TX\_TEST\_PACKET\_NUMBER output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-
Number_Of_Packets	4	Number of packets sent during the last direct TX test	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

### 2.3.6 ACI\_HAL\_TONE\_START

This command starts a carrier frequency (a tone) on a specific channel. The frequency sine wave at the specific channel can be used only for debugging purposes. The channel ID is a parameter from 0x00 to 0x27 for the 40 BLE channels (0x00 for 2.402 GHz, 0x01 for 2.404 GHz, etc). This command cannot be used when normal Bluetooth activities are ongoing. The tone is stopped by an ACI\_HAL\_TONE\_STOP command.

**Table 145. ACI\_HAL\_TONE\_START input parameters**

Parameter	Size	Description	Possible values
RF_Channel	1	BLE channel ID, from 0x00 to 0x27, meaning (2.402 + 0.002 * 0xXX) GHz. The device continuously emits 0s, indicating that the tone is at the channel center frequency less than the maximum frequency deviation (250 kHz).	0x00 ... 0x27
Freq_offset	1	Frequency offset for tone channel	0x00 ... 0xFF

**Table 146. ACI\_HAL\_TONE\_START output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

#### Events generated

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

### 2.3.7 ACI\_HAL\_TONE\_STOP

This command is used to stop the previously started ACI\_HAL\_TONE\_START command.

**Input parameters:** none

**Table 147. ACI\_HAL\_TONE\_STOP output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

#### Events generated

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

### 2.3.8 ACI\_HAL\_GET\_LINK\_STATUS

This command returns the status of the eight Bluetooth Low Energy links managed by the device.

**Input parameters:** none

**Table 148. ACI\_HAL\_GET\_LINK\_STATUS output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-
Link_Status	8	Array of link status (eight links). Each link status is 1 byte.	<ul style="list-style-type: none"> <li>• 0x00: Idle</li> <li>• 0x01: Advertising</li> <li>• 0x02: Connected in peripheral role</li> <li>• 0x03: Scanning</li> <li>• 0x04: Reserved</li> <li>• 0x05: Connected in central role</li> <li>• 0x06: TX test mode</li> <li>• 0x07: RX test mode</li> <li>• 0x81: Advertising with additional beacon</li> </ul>
Link_Connection_Handle	16	Array of connection handles (two bytes) for eight links. Valid only if the link status is "connected" (0x02 or 0x05).	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

### 2.3.9 ACI\_HAL\_SET\_RADIO\_ACTIVITY\_MASK

This command sets the bitmask associated to ACI\_HAL\_END\_OF\_RADIO\_ACTIVITY\_EVENT. Only the activities enabled in the mask are reported to application by ACI\_HAL\_END\_OF\_RADIO\_ACTIVITY\_EVENT.

**Table 149. ACI\_HAL\_SET\_RADIO\_ACTIVITY\_MASK input parameters**

Parameter	Size	Description	Possible values
Radio_Activity_Mask	2	Bitmask of radio events	<p>Bitmask of:</p> <ul style="list-style-type: none"> <li>• 0x0001: Idle</li> <li>• 0x0002: Advertising</li> <li>• 0x0004: Connection event peripheral</li> <li>• 0x0008: Scanning</li> <li>• 0x0010: Connection request</li> <li>• 0x0020: Connection event central</li> <li>• 0x0040: TX test mode</li> <li>• 0x0080: RX test mode</li> </ul>

**Table 150. ACI\_HAL\_SET\_RADIO\_ACTIVITY\_MASK output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)
- [ACI\\_HAL\\_END\\_OF\\_RADIO\\_ACTIVITY\\_EVENT](#)

### 2.3.10 ACI\_HAL\_GET\_ANCHOR\_PERIOD

This command returns information about the Anchor Period, to help an application operating in multi-link scenarios in the selection of timing slots.

**Input parameters:** none

**Table 151.** ACI\_HAL\_GET\_ANCHOR\_PERIOD output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-
Anchor_Period	4	Current anchor period. $T = N * 0.625 \text{ ms}$	-
Max_Free_Slot	4	Maximum available time that can be allocated for a new slot. $T = N * 0.625 \text{ ms}$	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.3.11 ACI\_HAL\_SET\_EVENT\_MASK**

This command is used to enable/disable the generation of HAL events. If the bit in the Event\_Mask is set to 1, the event associated with that bit is enabled.

**Table 152.** ACI\_HAL\_SET\_EVENT\_MASK input parameters

Parameter	Size	Description	Possible values
Event_Mask	4	Mask to enable/disable generation of HAL events	Bitmask of: <ul style="list-style-type: none"><li>• 0x00000000: No events specified (default)</li><li>• 0x00000001: ACI_HAL_SCAN_REQ_REPORT_EVENT</li></ul>

**Table 153.** ACI\_HAL\_SET\_EVENT\_MASK output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.3.12 ACI\_HAL\_GET\_PM\_DEBUG\_INFO**

This command is used to retrieve TX, RX, and total buffer count allocated for ACL packets.

**Input parameters:** none

**Table 154.** ACI\_HAL\_GET\_PM\_DEBUG\_INFO output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-
Allocated_For_TX	1	MBlocks allocated for TX	-
Allocated_For_RX	1	MBlocks allocated for RX	-
Allocated_MBlocks	1	Overall allocated MBlocks	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.3.13 ACI\_HAL\_SET\_PERIPHERAL\_LATENCY**

This command is used to disable/enable the peripheral latency feature (by default enabled) during a connection.

**Table 155.** ACI\_HAL\_SET\_PERIPHERAL\_LATENCY input parameters

Parameter	Size	Description	Possible values
Enable	1	Enable/disable peripheral latency	<ul style="list-style-type: none"><li>• 0x00: latency disabled</li><li>• 0x01: latency enabled</li></ul>

**Table 156.** ACI\_HAL\_SET\_PERIPHERAL\_LATENCY output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.3.14 ACI\_HAL\_READ\_RSSI**

This command returns the RSSI value.

**Input parameters:** none

**Table 157.** ACI\_HAL\_READ\_RSSI output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-
RSSI	1	RSSI (signed integer), in dBm	<ul style="list-style-type: none"><li>• 127: RSSI not available</li><li>• -127 ... 20</li></ul>

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.3.15 ACI\_HAL\_READ\_RADIO\_REG**

This command reads the register value from the RF module.

**Table 158.** ACI\_HAL\_READ\_RADIO\_REG input parameters

Parameter	Size	Description	Possible values
Register_Address	1	Address of the register to read	-

**Table 159.** ACI\_HAL\_READ\_RADIO\_REG output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-
reg_val	1	Register value	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.3.16 ACI\_HAL\_WRITE\_RADIO\_REG**

This command writes Register value to the RF module.

**Table 160.** ACI\_HAL\_WRITE\_RADIO\_REG input parameters

Parameter	Size	Description	Possible values
Register_Address	1	Address of the register to be written	-
Register_Value	1	Value to be written	-

**Table 161.** ACI\_HAL\_WRITE\_RADIO\_REG output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- HCI\_COMMAND\_COMPLETE\_EVENT

### 2.3.17 ACI\_HAL\_READ\_RAW\_RSSI

This command returns the raw value of the RSSI.

**Input parameters:** none

**Table 162.** ACI\_HAL\_READ\_RAW\_RSSI output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-
Value	3	Raw RSSI value	-

**Events generated**

- HCI\_COMMAND\_COMPLETE\_EVENT

### 2.3.18 ACI\_HAL\_RX\_START

This command sets up the RF to listen to a specific RF channel.

**Table 163.** ACI\_HAL\_RX\_START input parameters

Parameter	Size	Description	Possible values
RF_Channel	1	BLE channel ID, from 0x00 to 0x27 meaning $(2.402 + 0.002 * 0\text{XX}) \text{ GHz}$ . The device continuously emits 0s, meaning that the tone is at the channel centre frequency, minus the maximum frequency deviation (250 kHz).	0x00 ... 0x27

**Table 164.** ACI\_HAL\_RX\_START output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- HCI\_COMMAND\_COMPLETE\_EVENT

### 2.3.19 ACI\_HAL\_RX\_STOP

This command stops a previous ACI\_HAL\_RX\_START command.

**Input parameters:** none

**Table 165.** ACI\_HAL\_RX\_STOP output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

### 2.3.20 ACI\_HAL\_STACK\_RESET

This command is equivalent to HCI\_RESET, but ensures that Sleep mode is entered immediately after its completion.

**Input parameters:** none

**Table 166.** ACI\_HAL\_STACK\_RESET output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

## 2.4

### GAP commands

In Table 167 "Y" means that the corresponding command applies to the dedicated BLE stack variant, namely LO / BO / PO / BF / LB, respectively, for Link layer only / Beacon only / Peripheral only / Basic feature / Link layer only basic.

Table 167. GAP commands list

Command	OpCode	LO	PO	BO	BF	LB
ACI_GAP_SET_NON_DISCOVERABLE	0xFC81	-	Y	-	Y	-
ACI_GAP_SET_LIMITED_DISCOVERABLE	0xFC82	-	Y	-	Y	-
ACI_GAP_SET_DISCOVERABLE	0xFC83	-	Y	-	Y	-
ACI_GAP_SET_DIRECT_CONNECTABLE	0xFC84	-	Y	-	Y	-
ACI_GAP_SET_IO_CAPABILITY	0xFC85	-	Y	-	Y	-
ACI_GAP_SET_AUTHENTICATION_REQUIREMENT	0xFC86	-	Y	-	Y	-
ACI_GAP_SET_AUTHORIZATION_REQUIREMENT	0xFC87	-	Y	-	Y	-
ACI_GAP_PASS_KEY_RESP	0xFC88	-	Y	-	Y	-
ACI_GAP_AUTHORIZATION_RESP	0xFC89	-	Y	-	Y	-
ACI_GAP_INIT	0xFC8A	-	Y	-	Y	-
ACI_GAP_SET_NON_CONNECTABLE	0xFC8B	-	Y	-	Y	-
ACI_GAP_SET_UNDIRECTED_CONNECTABLE	0xFC8C	-	Y	-	Y	-
ACI_GAP_PERIPHERAL_SECURITY_REQ	0xFC8D	-	Y	-	Y	-
ACI_GAP_UPDATE_ADV_DATA	0xFC8E	-	Y	-	Y	-
ACI_GAP_DELETE_AD_TYPE	0xFC8F	-	Y	-	Y	-
ACI_GAP_GET_SECURITY_LEVEL	0xFC90	-	Y	-	Y	-
ACI_GAP_SET_EVENT_MASK	0xFC91	-	Y	-	Y	-
ACI_GAP_CONFIGURE_FILTER_ACCEPT_LIST	0xFC92	-	Y	-	Y	-
ACI_GAP_TERMINATE	0xFC93	-	Y	-	Y	-
ACI_GAP_CLEAR_SECURITY_DB	0xFC94	-	Y	-	Y	-
ACI_GAP_ALLOW_REBOND	0xFC95	-	Y	-	Y	-
ACI_GAP_START_LIMITED_DISCOVERY_PROC	0xFC96	-	-	-	Y	-
ACI_GAP_START_GENERAL_DISCOVERY_PROC	0xFC97	-	-	-	Y	-
ACI_GAP_START_AUTO_CONNECTION_ESTABLISH_PROC	0xFC99	-	-	-	Y	-
ACI_GAP_START_GENERAL_CONNECTION_ESTABLISH_PROC	0xFC9A	-	-	-	Y	-
ACI_GAP_START_SELECTIVE_CONNECTION_ESTABLISH_PROC	0xFC9B	-	-	-	Y	-
ACI_GAP_CREATE_CONNECTION	0xFC9C	-	-	-	Y	-
ACI_GAP_TERMINATE_GAP_PROC	0xFC9D	-	-	-	Y	-
ACI_GAP_START_CONNECTION_UPDATE	0xFC9E	-	-	-	Y	-
ACI_GAP_SEND_PAIRING_REQ	0xFC9F	-	-	-	Y	-
ACI_GAP_RESOLVE_PRIVATE_ADDR	0xFCA0	-	-	-	Y	-
ACI_GAP_SET_BROADCAST_MODE	0xFCA1	-	-	-	Y	-
ACI_GAP_START_OBSERVATION_PROC	0xFCA2	-	-	-	Y	-
ACI_GAP_GET_BONDED_DEVICES	0xFCA3	-	Y	-	Y	-
ACI_GAP_IS_DEVICE_BONDED	0xFCA4	-	Y	-	Y	-
ACI_GAP_NUMERIC_COMPARISON_VALUE_CONFIRM_YESNO	0xFCA5	-	Y	-	Y	-

Command	OpCode	LO	PO	BO	BF	LB
ACI_GAP_PASSKEY_INPUT	0xFCA6	-	Y	-	Y	-
ACI_GAP_GET_OOB_DATA	0xFCA7	-	Y	-	Y	-
ACI_GAP_SET_OOB_DATA	0xFCA8	-	Y	-	Y	-
ACI_GAP_ADD_DEVICES_TO_RESOLVING_LIST	0xFCA9	-	Y	-	Y	-
ACI_GAP_REMOVE_BONDED_DEVICE	0xFCAA	-	Y	-	Y	-
ACI_GAP_ADD_DEVICES_TO_LIST	0xFCAB	-	Y	-	Y	-
ACI_GAP_ADDITIONAL_BEACON_START	0xFCB0	-	Y	-	Y	-
ACI_GAP_ADDITIONAL_BEACON_STOP	0xFCB1	-	Y	-	Y	-
ACI_GAP_ADDITIONAL_BEACON_SET_DATA	0xFCB2	-	Y	-	Y	-
ACI_GAP_ADV_SET_CONFIGURATION	0xFCC0	-	-	-	-	-
ACI_GAP_ADV_SET_ENABLE	0xFCC1	-	-	-	-	-
ACI_GAP_ADV_SET_ADV_DATA	0xFCC2	-	-	-	-	-
ACI_GAP_ADV_SET_SCAN RESP DATA	0xFCC3	-	-	-	-	-
ACI_GAP_ADV_REMOVE_SET	0xFCC4	-	-	-	-	-
ACI_GAP_ADV_CLEAR_SETS	0xFCC5	-	-	-	-	-
ACI_GAP_ADV_SET_RANDOM_ADDRESS	0xFCC6	-	-	-	-	-
ACI_GAP_EXT_START_SCAN	0xFCD0	-	-	-	-	-
ACI_GAP_EXT_CREATE_CONNECTION	0xFCD1	-	-	-	-	-

#### 2.4.1 ACI\_GAP\_SET\_NON\_DISCOVERABLE

Puts the device in non-discoverable mode. This command, which supports only legacy advertising, disables the LL advertising.

**Input parameters:** none

**Table 168. ACI\_GAP\_SET\_NON\_DISCOVERABLE output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

##### Events generated

- HCI\_COMMAND\_COMPLETE\_EVENT

#### 2.4.2 ACI\_GAP\_SET\_LIMITED\_DISCOVERABLE

Puts the device in limited discoverable mode, making it discoverable for a maximum period of TGAP (lim\_adv\_timeout) = 180 seconds (from errata). The advertising can be disabled at any time by issuing the ACI\_GAP\_SET\_NON\_DISCOVERABLE command.

The Adv\_Interval\_Min and Adv\_Interval\_Max parameters are optional, if both are set to 0, the GAP uses default values for adv intervals for limited discoverable mode (250 and 500 ms, respectively). To allow a fast connection, the host can set Local\_Name, Service\_Uuid\_List, Conn\_Interval\_Min and Conn\_Interval\_Max. If provided, these data are inserted into the advertising packet payload as AD data: these parameters are optional, their values can be set in advertised data using the ACI\_GAP\_UPDATE\_ADV\_DATA command separately. The total size of data in advertising packet cannot exceed 31 bytes.

With this command, the BLE stack also adds automatically the following standard AD types:

- AD flags
- Power level when advertising timeout happens (i.e. limited discovery period has elapsed), controller generates an ACI\_GAP\_LIMITED\_DISCOVERABLE\_EVENT

**Table 169. ACI\_GAP\_SET\_LIMITED\_DISCOVERABLE input parameters**

Parameter	Size	Description	Possible values
Advertising_Type	1	Advertising type	<ul style="list-style-type: none"> <li>• 0x00: ADV_IND (connectable undirected advertising)</li> <li>• 0x02: ADV_SCAN_IND (scannable undirected advertising)</li> <li>• 0x03: ADV_NONCONN_IND (non connectable undirected advertising)</li> </ul>
Advertising_Interval_Min	2	Minimum advertising interval. Time = N * 0.625 ms	0x0020 (20 ms) ... 0x4000 (10240 ms)
Advertising_Interval_Max	2	Maximum advertising interval. Time = N * 0.625 ms	0x0020 (20 ms) ... 0x4000 (10240 ms)
Own_Address_Type	1	Own address type: if Privacy is disabled, the address can be public or static random, otherwise, it can be a resolvable or a non-resolvable private address.	<ul style="list-style-type: none"> <li>• 0x00: Public device address</li> <li>• 0x01: Random device address</li> <li>• 0x02: Resolvable private address</li> <li>• 0x03: Non resolvable private address</li> </ul>
Advertising_Filter_Policy	1	Advertising filter policy: not applicable (the value of Advertising_Filter_Policy parameter is not used inside the Stack).	-
Local_Name_Length	1	Length of the local_name field in octets. If length is set to 0x00, Local_Name parameter is not used.	-
Local_Name	Local_Name_Length	Local name of the device. First byte must be 0x08 for shortened local name, 0x09 for complete local name. No NULL character at the end.	-
Service_Uuid_length	1	Length of the service UUID list in octets. If there is no service to advertise, set this field to 0x00.	-
Service_Uuid_List	Service_Uuid_length	This is the list of the uuids. First byte is the AD Type.	-
Conn_Interval_Min	2	Peripheral connection interval minimum value suggested by peripheral. If Conn_Interval_Min and Conn_Interval_Max are not 0x0000, peripheral connection interval range AD structure is added in advertising data. Connection interval is defined as connIntervalmin = Conn_Interval_Min x 1.25 ms.	<ul style="list-style-type: none"> <li>• 0x0000 (NaN)</li> <li>• 0xFFFF (NaN): no specific minimum</li> <li>• 0x0006 (7.50 ms) ... 0x0C80 (4000 ms)</li> </ul>
Conn_Interval_Max	2	Peripheral connection interval maximum value suggested by peripheral. If Conn_Interval_Min and Conn_Interval_Max are not 0x0000, peripheral connection interval range AD structure is added in advertising data. Connection interval is defined as connIntervalmax = Conn_Interval_Max x 1.25 ms	<ul style="list-style-type: none"> <li>• 0x0000 (NaN)</li> <li>• 0xFFFF (NaN) : no specific maximum</li> <li>• 0x0006 (7.50 ms) ... 0x0C80 (4000 ms)</li> </ul>

**Table 170.** ACI\_GAP\_SET\_LIMITED\_DISCOVERABLE output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- HCI\_COMMAND\_COMPLETE\_EVENT
- ACI\_GAP\_LIMITED\_DISCOVERABLE\_EVENT

**2.4.3 ACI\_GAP\_SET\_DISCOVERABLE**

This command, which supports only legacy advertising, puts the device in general discoverable mode. The device is discoverable until the host issues the ACI\_GAP\_SET\_NON\_DISCOVERABLE command. The Adv\_Interval\_Min and Adv\_Interval\_Max parameters are optional, if both are set to 0, the GAP uses the default values for general discoverable mode:

- when using connectable undirected advertising events:
  - Adv\_Interval\_Min = 30 ms
  - Adv\_Interval\_Max = 60 ms
- when using non-connectable advertising events or scannable undirected advertising events:
  - Adv\_Interval\_Min = 100 ms
  - Adv\_Interval\_Max = 150 ms

Host can set the local name, a service UUID list, and the peripheral connection interval range. If provided, these data are inserted into the advertising packet payload as AD data. These parameters are optional in this command, their values can be also set using aci\_gap\_update\_adv\_data() separately. The total size of data in advertising packet cannot exceed 31 bytes. With this command, the BLE stack also adds automatically the following standard AD types:

- AD flags
- TX power level

**Table 171.** ACI\_GAP\_SET\_DISCOVERABLE input parameters

Parameter	Size	Description	Possible values
Advertising_Type	1	Advertising type.	<ul style="list-style-type: none"><li>• 0x00: ADV_IND (connectable undirected advertising)</li><li>• 0x02: ADV_SCAN_IND (scannable undirected advertising)</li><li>• 0x03: ADV_NONCONN_IND (non connectable undirected advertising)</li></ul>
Advertising_Interval_Min	2	Minimum advertising interval. Time = N * 0.625 ms	0x0020 (20 ms) ... 0x4000 (10240 ms)
Advertising_Interval_Max	2	Maximum advertising interval Time = N * 0.625 ms	0x0020 (20 ms) ... 0x4000 (10240 ms)
Own_Address_Type	1	Own address type: if Privacy is disabled, the address can be public or static random, otherwise, it can be a resolvable or a non-resolvable private address.	<ul style="list-style-type: none"><li>• 0x00: Public device address</li><li>• 0x01: Random device address</li><li>• 0x02: Resolvable private address</li><li>• 0x03: Non resolvable private address</li></ul>
Advertising_Filter_Policy	1	Advertising filter policy: not applicable (the value of Advertising_Filter_Policy parameter is not used inside the stack)	-

Parameter	Size	Description	Possible values
Local_Name_Length	1	Length of the local_name field in octets. If length is set to 0x00, Local_Name parameter is not used.	-
Local_Name	Local_Name_Length	Local name of the device. First byte must be 0x08 for shortened local name or 0x09 for complete local name. No NULL character at the end.	-
Service_Uuid_length	1	Length of the service UUID list in octets. If there is no service to be advertised, set this field to 0x00.	-
Service_Uuid_List	Service_Uuid_length	This is the list of the UUIDs . First byte is the AD type.	-
Conn_Interval_Min	2	Peripheral connection interval minimum value suggested by peripheral. If Conn_Interval_Min and Conn_Interval_Max are not 0x0000, peripheral connection interval range AD structure is added in advertising data. Connection interval is defined as: connIntervalmin = Conn_Interval_Min x 1.25 ms.	<ul style="list-style-type: none"> <li>• 0x0000 (NaN)</li> <li>• 0xFFFF (NaN): no specific minimum</li> <li>• 0x0006 (7.50 ms) ... 0x0C80 (4000 ms)</li> </ul>
Conn_Interval_Max	2	Peripheral connection interval maximum value suggested by peripheral. If Conn_Interval_Min and Conn_Interval_Max are not 0x0000, peripheral connection interval range AD structure is added in advertising data. Connection interval is defined as: connIntervalmax = Conn_Interval_Max x 1.25 ms.	<ul style="list-style-type: none"> <li>• 0x0000 (NaN)</li> <li>• 0xFFFF (NaN): no specific maximum</li> <li>• 0x0006 (7.50 ms) ... 0x0C80 (4000 ms)</li> </ul>

**Table 172. ACI\_GAP\_SET\_DISCOVERABLE output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

#### Events generated

- `HCI_COMMAND_COMPLETE_EVENT`

#### 2.4.4 ACI\_GAP\_SET\_DIRECT\_CONNECTABLE

This command supports only legacy advertising, and sets the device in direct connectable mode.

The device advertises using high or low duty cycle connectable directed advertising events. The address used in advertising packets is defined by the Own\_Address\_Type parameter, depending upon privacy (enabled or not).

When using high duty cycle connectable directed advertising events, the device stays in directed connectable mode only for 1.28 seconds. If no connection is established within this duration, it enters non discoverable mode, and advertising must be re-enabled explicitly. The controller generates

`HCI_LE_CONNECTION_COMPLETE_EVENT` with the status set to `HCI_ADVERTISING_TIMEOUT_ERR_CODE` if the connection is not established, `BLE_STATUS_SUCCESS` (0x00) if the connection is successfully established.

**Table 173. ACI\_GAP\_SET\_DIRECT\_CONNECTABLE input parameters**

Parameter	Size	Description	Possible values
Own_Address_Type	1	Own address type: if Privacy is disabled, the address can be public or static random, otherwise, it can be a resolvable private address.	<ul style="list-style-type: none"> <li>• 0x00: Public device address</li> <li>• 0x01: Random device address</li> <li>• 0x02: Resolvable private address</li> </ul>

Parameter	Size	Description	Possible values
Directed_Advertising_Type	1	Advertising type	<ul style="list-style-type: none"> <li>• 0x01: High duty cycle directed advertising</li> <li>• 0x04: Low duty cycle directed advertising</li> </ul>
Direct_Address_Type	1	The address type of the peer device.	<ul style="list-style-type: none"> <li>• 0x00: Public device address</li> <li>• 0x01: Random device address</li> </ul>
Direct_Address	6	Initiator Bluetooth address	-
Advertising_Interval_Min	2	Minimum advertising interval. Time = N * 0.625 ms	<ul style="list-style-type: none"> <li>• 0x0006 (3.75 ms): for high duty cycle directed advertising</li> <li>• 0x0020 (20 ms) ... 0x4000 (10240 ms): for low duty cycle directed advertising</li> </ul>
Advertising_Interval_Max	2	Maximum advertising interval. Time = N * 0.625 msec.	<ul style="list-style-type: none"> <li>• 0x0006 (3.75 ms) : for high duty cycle directed advertising</li> <li>• 0x0020 (20 ms) ... 0x4000 (10240 ms) : for low duty cycle directed advertising</li> </ul>

**Table 174. ACI\_GAP\_SET\_DIRECT\_CONNECTABLE output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

#### Events generated

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)
- [HCI\\_LE\\_CONNECTION\\_COMPLETE\\_EVENT](#)

### 2.4.5 ACI\_GAP\_SET\_IO\_CAPABILITY

This command set the device IO capabilities. It must be given only when the device is not in a connected state.

**Table 175. ACI\_GAP\_SET\_IO\_CAPABILITY input parameters**

Parameter	Size	Description	Possible values
IO_Capability	1	IO capability of the device	<ul style="list-style-type: none"> <li>• 0x00: IO_CAP_DISPLAY_ONLY</li> <li>• 0x01: IO_CAP_DISPLAY_YES_NO</li> <li>• 0x02: IO_CAP_KEYBOARD_ONLY</li> <li>• 0x03: IO_CAP_NO_INPUT_NO_OUTPUT</li> <li>• 0x04: IO_CAP_KEYBOARD_DISPLAY</li> </ul>

**Table 176. ACI\_GAP\_SET\_IO\_CAPABILITY output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

#### Events generated

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

### 2.4.6 ACI\_GAP\_SET\_AUTHENTICATION\_REQUIREMENT

Sets the authentication requirements for the device. If the OOB\_Enable is set to 0, the following 16 octets of OOB\_Data are ignored on reception. This command must be given only when the device is not in a connected state.

**Table 177. ACI\_GAP\_SET\_AUTHENTICATION\_REQUIREMENT input parameters**

Parameter	Size	Description	Possible values
Bonding_Mode	1	Bonding mode. Only if bonding is enabled (0x01), the bonding information is stored in flash	<ul style="list-style-type: none"> <li>• 0x00: No-bonding mode</li> <li>• 0x01: Bonding mode</li> </ul>
MITM_Mode	1	MITM mode	<ul style="list-style-type: none"> <li>• 0x00: MITM protection not required</li> <li>• 0x01: MITM protection required</li> </ul>
SC_Support	1	LE secure connections support	<ul style="list-style-type: none"> <li>• 0x00: Secure connections pairing not supported</li> <li>• 0x01: Secure connections pairing supported but optional</li> <li>• 0x02: Secure connections pairing supported and mandatory (SC only mode)</li> </ul>
KeyPress_Notification_Support	1	Keypress notification support	<ul style="list-style-type: none"> <li>• 0x00: Keypress notification not supported</li> <li>• 0x01: Keypress notification supported</li> </ul>
Min_Encryption_Key_Size	1	Minimum encryption key size to be used during pairing	-
Max_Encryption_Key_Size	1	Maximum encryption key size to use during pairing	-
Use_Fixed_Pin	1	Use or not fixed pin. If set to 0x00, during the pairing process the application is not requested for a pin (Fixed_Pin is used). If set to 0x01, during pairing process, if a passkey is required the application is notified.	<ul style="list-style-type: none"> <li>• 0x00: use a fixed pin</li> <li>• 0x01: do not use a fixed pin</li> </ul>
Fixed_Pin	4	Fixed pin to use during pairing if MITM protection is enabled (random value)	0 ... 999999
Identity_Address_Type	1	Identity address type	<ul style="list-style-type: none"> <li>• 0x00: Public identity address</li> <li>• 0x01: Random (static) identity address</li> </ul>

**Table 178. ACI\_GAP\_SET\_AUTHENTICATION\_REQUIREMENT output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

#### 2.4.7 ACI\_GAP\_SET\_AUTHORIZATION\_REQUIREMENT

Sets the authorization requirements of the device. This command has to be given when connected to a device if authorization is required to access services which require authorization.

**Table 179. ACI\_GAP\_SET\_AUTHORIZATION\_REQUIREMENT input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle for which the command is given.	0x0000 ... 0x0EFF
Authorization_Enable	1	Enable the authorization in the device and when a remote device tries to read/write a characteristic with authorization requirements, the stack sends back an error response with "Insufficient authorization" error code. After pairing is complete a ACI_GAP_AUTHORIZATION_REQ_EVENT is sent to the host.	<ul style="list-style-type: none"> <li>• 0x00: Authorization not required</li> <li>• 0x01: Authorization required</li> </ul>

**Table 180. ACI\_GAP\_SET\_AUTHORIZATION\_REQUIREMENT output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

## 2.4.8 ACI\_GAP\_PASS\_KEY\_RESP

This command is sent by the host in response to an ACI\_GAP\_PASS\_KEY\_REQ\_EVENT. The command parameters contain the pass key used during the pairing process.

**Table 181. ACI\_GAP\_PASS\_KEY\_RESP input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle for which the command is given.	0x0000 ... 0x0EFF
Pass_Key	4	Pass key used during the pairing process. Must be a six-digit decimal number.	0 ... 999999

**Table 182. ACI\_GAP\_PASS\_KEY\_RESP output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

## 2.4.9 ACI\_GAP\_AUTHORIZATION\_RESP

This command is sent by the host in response to an ACI\_GAP\_AUTHORIZATION\_REQ\_EVENT, to authorize a device to access attributes.

**Table 183. ACI\_GAP\_AUTHORIZATION\_RESP input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle for which the command is given	0x0000 ... 0x0EFF
Authorize	1	Authorization response	<ul style="list-style-type: none"> <li>• 0x01: Authorize</li> <li>• 0x02: Reject</li> </ul>

**Table 184. ACI\_GAP\_AUTHORIZATION\_RESP output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

#### 2.4.10 ACI\_GAP\_INIT

Initializes the GAP layer, and registers the GAP service with the GATT. All the standard GAP characteristics are added:

- device name
- appearance
- peripheral preferred connection parameters (peripheral role only): if added, its handle is equal to the Appearance characteristic handle plus 2

If privacy is enabled, the command unmasks the [HCI\\_LE\\_ENHANCED\\_CONNECTION\\_COMPLETE\\_EVENT](#).

**Table 185. ACI\_GAP\_INIT input parameters**

Parameter	Size	Description	Possible values
Role	1	Bitmap of allowed roles	Bitmask of: <ul style="list-style-type: none"><li>• 0x01: Peripheral</li><li>• 0x02: Broadcaster</li><li>• 0x04: Central</li><li>• 0x08: Observer</li></ul>
privacy_enabled	1	Specifies if privacy is enabled or not (only Controller privacy is supported)	<ul style="list-style-type: none"><li>• 0x00: Privacy disabled</li><li>• 0x02: Privacy enabled</li></ul>
device_name_char_len	1	Length of the device name characteristic	-

**Table 186. ACI\_GAP\_INIT output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-
Service_Handle	2	Handle of the GAP service	-
Dev_Name_Char_Handle	2	Device name characteristic handle	-
Appearance_Char_Handle	2	Appearance characteristic handle	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

#### 2.4.11 ACI\_GAP\_SET\_NON\_CONNECTABLE

Puts the device into non connectable mode, not supporting connections. The privacy setting done in the ACI\_GAP\_INIT command plays a role in deciding the valid parameters for this command. Advertiser filter policy is internally set to 0x00.

**Table 187. ACI\_GAP\_SET\_NON\_CONNECTABLE input parameters**

Parameter	Size	Description	Possible values
Advertising_Event_Type	1	Advertising type.	<ul style="list-style-type: none"><li>• 0x02: ADV_SCAN_IND (scannable undirected advertising)</li><li>• 0x03: ADV_NONCONN_IND (non connectable undirected advertising)</li></ul>
Own_Address_Type	1	Own address type: if Privacy is disabled, the address can be public or static random, otherwise, it can be a resolvable or a non resolvable private address.	<ul style="list-style-type: none"><li>• 0x00: Public device address</li><li>• 0x01: Random device address</li><li>• 0x02: Resolvable private address</li><li>• 0x03: Non resolvable private address</li></ul>

**Table 188. ACI\_GAP\_SET\_NON\_CONNECTABLE output parameters**

Parameter	Size	Description	Possible values
Status	1	Status Error code	-

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

#### 2.4.12 ACI\_GAP\_SET\_UNDIRECTED\_CONNECTABLE

This command, which supports only legacy advertising, puts the device into undirected connectable mode. If privacy is enabled, a resolvable private address is generated and used as the advertiser's address. If not, the address of the type specified in own\_addr\_type is used for advertising.

**Table 189. ACI\_GAP\_SET\_UNDIRECTED\_CONNECTABLE input parameters**

Parameter	Size	Description	Possible values
Advertising_Interval_Min	2	Minimum advertising interval. Time = N * 0.625 ms	0x0020 (20 ms) ... 0x4000 (10240 ms)
Advertising_Interval_Max	2	Maximum advertising interval. Time = N * 0.625 ms	0x0020 (20 ms) ... 0x4000 (10240 ms)
Own_Address_Type	1	Own address type: if privacy is disabled, the address can be public or static random, otherwise, it can be a resolvable private address.	<ul style="list-style-type: none"> <li>• 0x00: Public device address</li> <li>• 0x01: Random device address</li> <li>• 0x02: Resolvable private address</li> </ul>
Adv_Filter_Policy	1	Advertising filter policy	<ul style="list-style-type: none"> <li>• 0x00: Allow scan and connect requests from any</li> <li>• 0x03: Allow scan and connect requests only from white list</li> </ul>

**Table 190. ACI\_GAP\_SET\_UNDIRECTED\_CONNECTABLE output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

#### Events generated

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

#### 2.4.13 ACI\_GAP\_PERIPHERAL\_SECURITY\_REQ

This command sends a security request to the central, to notify its security requirements. The central can encrypt the link, initiate the pairing procedure, or reject the request.

**Table 191. ACI\_GAP\_PERIPHERAL\_SECURITY\_REQ input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle for which the command is given.	0x0000 ... 0x0EFF

**Table 192. ACI\_GAP\_PERIPHERAL\_SECURITY\_REQ output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

#### Events generated

- [HCI\\_COMMAND\\_STATUS\\_EVENT](#)

- ACI\_GAP\_PERIPHERAL\_SECURITY\_INITIATED\_EVENT

#### 2.4.14 ACI\_GAP\_UPDATE\_ADV\_DATA

This command can be used to update the advertising data for a particular AD type. If the AD type specified does not exist, then it is added to the advertising data. If the overall advertising data length is more than 31 octets after the update, then the command is rejected and the old data is retained.

**Table 193. ACI\_GAP\_UPDATE\_ADV\_DATA input parameters**

Parameter	Size	Description	Possible values
AdvDataLen	1	Length of AdvData in octets	-
AdvData	AdvDataLen	Advertising data used by the device while advertising.	-

#### Output parameters

**Table 194. ACI\_GAP\_UPDATE\_ADV\_DATA output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

#### Events generated

- HCI\_COMMAND\_COMPLETE\_EVENT

#### 2.4.15 ACI\_GAP\_DELETE\_AD\_TYPE

This command can be used to delete the specified AD type from the advertisement data if present.

**Table 195. ACI\_GAP\_DELETE\_AD\_TYPE input parameters**

Parameter	Size	Description	Possible values
ADType	1	One of the AD types like in Bluetooth specification	-

**Table 196. ACI\_GAP\_DELETE\_AD\_TYPE output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

#### Events generated

- HCI\_COMMAND\_COMPLETE\_EVENT

#### 2.4.16 ACI\_GAP\_GET\_SECURITY\_LEVEL

This command can be used to get the current security settings of the device.

**Table 197. ACI\_GAP\_GET\_SECURITY\_LEVEL input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle for which the command is given	0x0000 ... 0x0EFF

**Table 198. ACI\_GAP\_GET\_SECURITY\_LEVEL output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-
Security_Mode	1	Security mode	<ul style="list-style-type: none"> <li>• 0x01: Security mode 1</li> <li>• 0x02: Security mode 2</li> </ul>
Security_Level	1	Security level	<ul style="list-style-type: none"> <li>• 0x01: Security level 1</li> <li>• 0x02: Security level 2</li> <li>• 0x03: Security level 3</li> <li>• 0x04: Security level 4</li> </ul>

**Events generated**

- **HCI\_COMMAND\_COMPLETE\_EVENT**

#### 2.4.17 ACI\_GAP\_SET\_EVENT\_MASK

This command makes it possible to mask events from the GAP. If the bit in the GAP\_Evt\_Mask is set to 1, the event associated with that bit is enabled. The default configuration is all the events masked.

**Table 199. ACI\_GAP\_SET\_EVENT\_MASK input parameters**

Parameter	Size	Description	Possible values
GAP_Evt_Mask	2	GAP event mask (default: 0xFFFF)	<p>Bitmask of:</p> <ul style="list-style-type: none"> <li>• 0x0000: No events</li> <li>• 0x0001: ACI_GAP_LIMITED_DISCOVERABLE_EVENT</li> <li>• 0x0002: ACI_GAP_PAIRING_COMPLETE_EVENT</li> <li>• 0x0004: ACI_GAP_PASS_KEY_REQ_EVENT</li> <li>• 0x0008: ACI_GAP_AUTHORIZATION_REQ_EVENT</li> <li>• 0x0010: ACI_GAP_PERIPHERAL_SECURITY_INITIATED_EVENT</li> <li>• 0x0020: ACI_GAP_BOND_LOST_EVENT</li> <li>• 0x0080: ACI_GAP_PROC_COMPLETE_EVENT</li> <li>• 0x0100: ACI_L2CAP_CONNECTION_UPDATE_REQ_EVENT</li> <li>• 0x0200: ACI_L2CAP_CONNECTION_UPDATE_RESP_EVENT</li> <li>• 0x0400: ACI_L2CAP_PROC_TIMEOUT_EVENT</li> <li>• 0x0800: ACI_GAP_ADDR_NOT_RESOLVED_EVENT</li> </ul>

**Table 200. ACI\_GAP\_SET\_EVENT\_MASK output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- **HCI\_COMMAND\_COMPLETE\_EVENT**

#### 2.4.18 ACI\_GAP\_CONFIGURE\_FILTER\_ACCEPT\_LIST

Add addresses of bonded devices into the controller's white list. The command returns an error if unable to add the bonded devices into the list.

**Input parameters:** none

**Table 201. ACI\_GAP\_CONFIGURE\_FILTER\_ACCEPT\_LIST output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.4.19 ACI\_GAP\_TERMINATE**

Command the controller to terminate the connection. An [HCI\\_DISCONNECTION\\_COMPLETE\\_EVENT](#) is generated when the link is disconnected. It is important to leave a 100 ms blank window before sending any new command (including system hardware reset), since immediately after [HCI\\_DISCONNECTION\\_COMPLETE\\_EVENT](#), the system saves important information in the non volatile memory.

**Table 202. ACI\_GAP\_TERMINATE input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle for which the command is given	0x0000 ... 0x0EFF
Reason	1	The reason for ending the connection.	<ul style="list-style-type: none"><li>• 0x05: Authentication failure</li><li>• 0x13: Remote user terminated connection</li><li>• 0x14: Remote device terminated connection due to low resources</li><li>• 0x15: Remote device terminated connection due to power-off</li><li>• 0x1A: Unsupported remote feature</li><li>• 0x3B: Unacceptable connection parameters</li></ul>

**Table 203. ACI\_GAP\_TERMINATE output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_STATUS\\_EVENT](#)
- [HCI\\_DISCONNECTION\\_COMPLETE\\_EVENT](#)

**2.4.20 ACI\_GAP\_CLEAR\_SECURITY\_DB**

Clears the bonding table, all the devices are removed from it. See [ACI\\_GAP\\_REMOVE\\_BONDED\\_DEVICE](#) to remove only one device.

**Note:** As a fallback mode, if the bonding table is full, the BLE stack automatically clears it before writing information about a new bonded device.

**Input parameters:** none

**Table 204. ACI\_GAP\_CLEAR\_SECURITY\_DB output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.4.21 ACI\_GAP\_ALLOW\_REBOND**

Allows the security manager to complete the pairing procedure and re-bond with the central. This command is given by the application when it receives the [ACI\\_GAP\\_BOND\\_LOST\\_EVENT](#), if it wants the re-bonding to happen successfully. If this command is not given on receiving the event, the bonding procedure times out.

**Table 205. ACI\_GAP\_ALLOW\_REBOND input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle for which the command is given	0x0000 ... 0x0EFF

**Table 206. ACI\_GAP\_ALLOW\_REBOND output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

#### 2.4.22 ACI\_GAP\_START\_LIMITED\_DISCOVERY\_PROC

Starts the limited discovery procedure. The controller starts active scanning, only the devices in limited discoverable mode are returned to the upper layers. The procedure is terminated when either the upper layers issue the command ACI\_GAP\_TERMINATE\_GAP\_PROC with the procedure code set to 0x01, or a there is a timeout (value fixed at 10.24 s). When the procedure is terminated, ACI\_GAP\_PROC\_COMPLETE\_EVENT is returned with the procedure code set to 0x01. The device found when the procedure is ongoing is returned to the upper layers through HCI\_LE\_ADVERTISING\_REPORT\_EVENT.

**Table 207. ACI\_GAP\_START\_LIMITED\_DISCOVERY\_PROC input parameters**

Parameter	Size	Description	Possible values
LE_Scan_Interval	2	The time interval from when the controller started its last LE scan, until it begins the subsequent one.  Time = N * 0.625 ms	0x0004 (2.50 ms) ... 0x4000 (10240 ms)
LE_Scan_Window	2	Amount of time for the duration of the LE scan. LE_Scan_Window is less than or equal to LE_Scan_Interval.  Time = N * 0.625 ms	0x0004 (2.50 ms) ... 0x4000 (10240 ms)
Own_Address_Type	1	Own address type: if Privacy is disabled, the address can be public or static random, otherwise, it can be a resolvable or a non-resolvable private address.	<ul style="list-style-type: none"> <li>• 0x00: Public device address</li> <li>• 0x01: Random device address</li> <li>• 0x02: Resolvable private address</li> <li>• 0x03: Non resolvable private address</li> </ul>
Filter_Duplicates	1	Enable/disable duplicate filtering.	<ul style="list-style-type: none"> <li>• 0x00: Duplicate filtering disabled</li> <li>• 0x01: Duplicate filtering enabled</li> </ul>

**Output parameters**
**Table 208. ACI\_GAP\_START\_LIMITED\_DISCOVERY\_PROC output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_STATUS\\_EVENT](#)
- [HCI\\_LE\\_ADVERTISING\\_REPORT\\_EVENT](#)
- [ACI\\_GAP\\_PROC\\_COMPLETE\\_EVENT](#)

## 2.4.23 ACI\_GAP\_START\_GENERAL\_DISCOVERY\_PROC

Starts the general discovery procedure. The controller is commanded to start active scanning. The procedure is terminated when either the upper layers issue the command ACI\_GAP\_TERMINATE\_GAP\_PROC with the procedure code set to 0x02, or when a timeout happens (value fixed at 10.24 s). When the procedure is terminated, ACI\_GAP\_PROC\_COMPLETE\_EVENT is returned with the procedure code set to 0x02.

The device found when the procedure is ongoing is returned to HCI\_LE\_ADVERTISING\_REPORT\_EVENT.

**Table 209. ACI\_GAP\_START\_GENERAL\_DISCOVERY\_PROC input parameters**

Parameter	Size	Description	Possible values
LE_Scan_Interval	2	Time interval from when the controller started its last LE scan, until it begins the subsequent one.  Time = N * 0.625 ms	<ul style="list-style-type: none"> <li>• 0x0004 (2.5 ms) ... 0x4000 (10240 ms): legacy advertising</li> <li>• 0x0004 (2.5 ms) ... 0x5DC0 (15000 ms): extended advertising</li> </ul>
LE_Scan_Window	2	Amount of time for the duration of the LE scan. LE_Scan_Window is less than or equal to LE_Scan_Interval.  Time = N * 0.625 ms	<ul style="list-style-type: none"> <li>• 0x0004 (2.5 ms) ... 0x4000 (10240 ms): legacy advertising</li> <li>• 0x0004 (2.5 ms) ... 0x5DC0 (15000 ms): extended advertising</li> </ul>
Own_Address_Type	1	Own address type: if Privacy is disabled, the address can be public or static random, otherwise, it can be a resolvable or a non-resolvable private address.	<ul style="list-style-type: none"> <li>• 0x00: Public device address</li> <li>• 0x01: Random device address</li> <li>• 0x02: Resolvable private address</li> <li>• 0x03: Non resolvable private address</li> </ul>
Filter_Duplicates	1	Enable/disable duplicate filtering	<ul style="list-style-type: none"> <li>• 0x00: Duplicate filtering disabled</li> <li>• 0x01: Duplicate filtering enabled</li> </ul>

**Table 210. ACI\_GAP\_START\_GENERAL\_DISCOVERY\_PROC output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

### Events generated

- [HCI\\_COMMAND\\_STATUS\\_EVENT](#)
- [HCI\\_LE\\_ADVERTISING\\_REPORT\\_EVENT](#)
- [ACI\\_GAP\\_PROC\\_COMPLETE\\_EVENT](#)

## 2.4.24 ACI\_GAP\_START\_AUTO\_CONNECTION\_ESTABLISH\_PROC

Starts the auto connection establishment procedure. The specified devices are added to the white list of the controller, and an LE\_Create\_Connection call is made to the controller by GAP with the initiator filter policy set to "use whitelist to determine which advertiser to connect to". When a command is issued to terminate the procedure by upper layer, a LE\_Create\_Connection\_Cancel call is made to the controller by GAP. The procedure is terminated when either a connection is successfully established with one of the specified devices in the white list, or when it is explicitly terminated by issuing the command ACI\_GAP\_TERMINATE\_GAP\_PROC with the procedure code set to 0x08. An ACI\_GAP\_PROC\_COMPLETE\_EVENT is returned with the procedure code set to 0x08. If controller privacy is enabled and the peer device (advertiser) is in the resolving list, the link layer generates an RPA, if not, the RPA/NRPA generated by the host is used.

**Table 211. ACI\_GAP\_START\_AUTO\_CONNECTION\_ESTABLISH\_PROC input parameters**

Parameter	Size	Description	Possible values
LE_Scan_Interval	2	Time interval from when the controller started its last LE scan until it begins the subsequent one.  Time = N * 0.625 ms	<ul style="list-style-type: none"> <li>• 0x0004 (2.5 ms) ... 0x4000 (10240 ms): legacy advertising</li> <li>• 0x0004 (2.5 ms) ... 0x5DC0 (15000 ms): extended advertising</li> </ul>

Parameter	Size	Description	Possible values
LE_Scan_Window	2	Amount of time for the duration of the LE scan. LE_Scan_Window is less than or equal to LE_Scan_Interval.  Time = N * 0.625 ms	<ul style="list-style-type: none"> <li>• 0x0004 (2.5 ms) ... 0x4000 (10240 ms): legacy advertising</li> <li>• 0x0004 (2.5 ms) ... 0x5DC0 (15000 ms): extended advertising</li> </ul>
Own_Address_Type	1	Own address type: if Privacy is disabled, the address can be public or static random, otherwise, it can be a resolvable or a non-resolvable private address.	<ul style="list-style-type: none"> <li>• 0x00: Public address</li> <li>• 0x01: Static random address</li> <li>• 0x02: Resolvable private address</li> <li>• 0x03: Non resolvable private address</li> </ul>
Conn_Interval_Min	2	Minimum value for the connection event interval. This is less than or equal to Conn_Interval_Max.  Time = N * 1.25 ms	0x0006 (7.5 ms) ... 0x0C80 (4000 ms)
Conn_Interval_Max	2	Maximum value for the connection event interval. This is greater than or equal to Conn_Interval_Min.  Time = N * 1.25 ms	0x0006 (7.5 ms) ... 0x0C80 (4000.00 ms)
Conn_Latency	2	Peripheral latency for the connection, in number of connection events.	0x0000 ... 0x01F3
Supervision_Timeout	2	Supervision timeout for the LE Link. It is a multiple of 10 ms and larger than $(1 + \text{connPeripheralLatency}) * \text{connInterval} * 2$ .  Time = N * 10 ms	0x000A (100 ms) ... 0x0C80 (32000 ms)
Minimum_CE_Length	2	Information parameter about the minimum length of connection needed for this LE connection.  Time = N * 0.625 ms	0x0000 (0.000 ms) ... 0xFFFF (40959.375 ms)
Maximum_CE_Length	2	Information parameter about the maximum length of connection needed for this LE connection.  Time = N * 0.625 ms	0x0000 (0.000 ms) ... 0xFFFF (40959.375 ms)
Num_of_Whitelist_Entries	1	Number of devices that have to be added to the white list.	-
Peer_Address_Type[i]	1	Address type	<ul style="list-style-type: none"> <li>• 0x00: Public address</li> <li>• 0x01: Random address</li> </ul>
Peer_Address[i]	6	Public or random address of the device to be added to the white list.	-

**Table 212. ACI\_GAP\_START\_AUTO\_CONNECTION\_ESTABLISH\_PROC output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

#### Events generated

- [HCI\\_COMMAND\\_STATUS\\_EVENT](#)
- [ACI\\_GAP\\_PROC\\_COMPLETE\\_EVENT](#)

## 2.4.25

**ACI\_GAP\_START\_GENERAL\_CONNECTION\_ESTABLISH\_PROC**

Starts a general connection establishment procedure. The host enables scanning in the controller with policy set to "accept all advertising packets" and from the scanning results, all the devices are sent to the upper layer using HCI\_LE\_ADVERTISING\_REPORT\_EVENT. The upper layer then selects the device to whom it wants to connect, by issuing the command ACI\_GAP\_CREATE\_CONNECTION. If privacy is enabled, a private resolvable or non resolvable address (based on the type specified in the command) is set as the scanner address. The ACI\_GAP\_CREATE\_CONNECTION command always uses a private resolvable address if the general connection establishment procedure is active. The procedure is terminated when a connection is established, or the upper layer terminates it by issuing the command ACI\_GAP\_TERMINATE\_GAP\_PROC with the procedure code set to 0x10. On procedure completion, an ACI\_GAP\_PROC\_COMPLETE\_EVENT is generated with code set to 0x10. If controller privacy is enabled and the peer device (advertiser) is in the resolving list, the link layer generates an RPA, if not, the RPA/NRPA generated by the host is used.

**Table 213. ACI\_GAP\_START\_GENERAL\_CONNECTION\_ESTABLISH\_PROC input parameters**

Parameter	Size	Description	Possible values
LE_Scan_Type	1	Passive or active scanning. With active scanning SCAN_REQ packets are sent.	<ul style="list-style-type: none"><li>• 0x00: Passive scanning</li><li>• 0x01: Active scanning</li></ul>
LE_Scan_Interval	2	Time interval from when the controller started its last LE scan, until it begins the subsequent one.  Time = N * 0.625 ms	<ul style="list-style-type: none"><li>• 0x0004 (2.5 ms) ... 0x4000 (10240 ms): legacy advertising</li><li>• 0x0004 (2.5 ms) ... 0x5DC0 (15000 ms): extended advertising</li></ul>
LE_Scan_Window	2	Amount of time for the duration of the LE scan. LE_Scan_Window is less than or equal to LE_Scan_Interval.  Time = N * 0.625 ms	<ul style="list-style-type: none"><li>• 0x0004 (2.5 ms) ... 0x4000 (10240 ms): legacy advertising</li><li>• 0x0004 (2.5 ms) ... 0x5DC0 (15000 ms): extended advertising</li></ul>
Own_Address_Type	1	Own address type: if Privacy is disabled, the address can be public or static random, otherwise, it can be a resolvable or a non-resolvable private address.	<ul style="list-style-type: none"><li>• 0x00: Public device address</li><li>• 0x01: Random device address</li><li>• 0x02: Resolvable private address</li><li>• 0x03: Non resolvable private address</li></ul>
Scanning_Filter_Policy	1	Scanning filter policy: <ul style="list-style-type: none"><li>• 0x00 Accept all advertisement packets. Directed packets not addressed to this device are ignored.</li><li>• 0x01 Ignore advertisement packets from devices not in the White list. Directed packets not addressed to this device are ignored.</li><li>• 0x02 Accept all undirected advertisement packets (allowed only if controller privacy is enabled). Directed packets where initiator address is an RPA and directed packets addressed to this device accepted.</li><li>• 0x03 Accept all undirected advertisement packets from devices in the White list. Directed packets where initiator address is an RPA and directed packets addressed to this device are accepted.</li></ul> If controller privacy is enabled Scanning_Filter_Policy can be only 0x00 or 0x02.	<ul style="list-style-type: none"><li>• 0x00: Accept all</li><li>• 0x01: Ignore devices not in the White List</li><li>• 0x02: Accept all (use resolving list)</li><li>• 0x03: Ignore devices not in the White list (use resolving list)</li></ul>
Filter_Duplicates	1	Enable/disable duplicate filtering.	<ul style="list-style-type: none"><li>• 0x00: Duplicate filtering disabled</li><li>• 0x01: Duplicate filtering enabled</li></ul>

**Table 214. ACI\_GAP\_START\_GENERAL\_CONNECTION\_ESTABLISH\_PROC output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_STATUS\\_EVENT](#)
- [ACI\\_GAP\\_PROC\\_COMPLETE\\_EVENT](#)

**2.4.26**
**ACI\_GAP\_START\_SELECTIVE\_CONNECTION\_ESTABLISH\_PROC**

Starts a selective connection establishment procedure.

The GAP adds the specified device addresses in the white list, and enables scanning in the controller with the scanner filter policy set to "accept packets only from devices in white list". All the found devices are sent to the upper layer by the [HCI\\_LE\\_ADVERTISING\\_REPORT\\_EVENT](#). The upper layer then selects the device to whom it wants to connect, by issuing the command [ACI\\_GAP\\_CREATE\\_CONNECTION](#). On completion of the procedure, an [ACI\\_GAP\\_PROC\\_COMPLETE\\_EVENT](#) generated, with the procedure code set to 0x20. The procedure is terminated when a connection is established, or when the upper layer terminates it by issuing the command [ACI\\_GAP\\_TERMINATE\\_GAP\\_PROC](#) with the procedure code set to 0x20. If controller privacy is enabled and the peer device (advertiser) is in the resolving list, the link layer generates a RPA, if not the RPA/NRPA generated by the host is used.

**Table 215. ACI\_GAP\_START\_SELECTIVE\_CONNECTION\_ESTABLISH\_PROC input parameters**

Parameter	Size	Description	Possible values
LE_Scan_Type	1	Passive or active scanning. With active scanning SCAN_REQ packets are sent.	<ul style="list-style-type: none"> <li>• 0x00: Passive scanning</li> <li>• 0x01: Active scanning</li> </ul>
LE_Scan_Interval	2	Time interval from when the controller started its last LE scan until it begins the subsequent LE scan.  Time = N * 0.625 ms	<ul style="list-style-type: none"> <li>• 0x0004 (2.5 ms) ... 0x4000 (10240 ms): legacy advertising</li> <li>• 0x0004 (2.5 ms) ... 0x5DC0 (15000 ms): extended advertising</li> </ul>
LE_Scan_Window	2	Amount of time for the duration of the LE scan. LE_Scan_Window is less than or equal to LE_Scan_Interval.  Time = N * 0.625 ms	<ul style="list-style-type: none"> <li>• 0x0004 (2.5 ms) ... 0x4000 (10240 ms): legacy advertising</li> <li>• 0x0004 (2.5 ms) ... 0x5DC0 (15000 ms): extended advertising</li> </ul>
Own_Address_Type	1	Own address type: if Privacy is disabled, the address can be public or static random, otherwise, it can be a resolvable or a non-resolvable private address.	<ul style="list-style-type: none"> <li>• 0x00: Public device address</li> <li>• 0x01: Random device address</li> <li>• 0x02: Resolvable private address</li> <li>• 0x03: Non resolvable private address</li> </ul>
Scanning_Filter_Policy	1	Scanning filter policy: <ul style="list-style-type: none"> <li>• 0x00 Accept all advertisement packets. Directed advertising packets not addressed to this device are ignored.</li> <li>• 0x01 Ignore advertisement packets from devices not in the white list. Directed advertising packets not addressed to this device are ignored.</li> </ul>	<ul style="list-style-type: none"> <li>• 0x00: Accept all</li> <li>• 0x01: Ignore devices not in the white list</li> <li>• 0x02: Accept all (use resolving list)</li> <li>• 0x03: Ignore devices not in the white list (use resolving list)</li> </ul>

Parameter	Size	Description	Possible values
		<ul style="list-style-type: none"> <li>• 0x02 Accept all undirected advertisement packets (allowed only if controller privacy is enabled). Directed advertisement packets where initiator address is an RPA and directed advertisement packets addressed to this device are accepted.</li> <li>• 0x03 Accept all undirected advertisement packets from devices that are in the white list. Directed advertisement packets where initiator address is RPA and directed advertisement packets addressed to this device is accepted.</li> </ul> <p><i>Note:</i> <i>if controller privacy is enabled Scanning_Filter_Policy can be only 0x01 or 0x03.</i></p>	
Filter_Duplicates	1	Enable/disable duplicate filtering.	<ul style="list-style-type: none"> <li>• 0x00: Duplicate filtering disabled</li> <li>• 0x01: Duplicate filtering enabled</li> </ul>
Num_of_Whitelist_Entries	1	Number of devices to add to the white list.	-
Peer_Address_Type[i]	1	Address type.	<ul style="list-style-type: none"> <li>• 0x00: Public device address</li> <li>• 0x01: Random device address</li> </ul>
Peer_Address[i]	6	Public or random address of the device to be added to the white list.	-

**Table 216. ACI\_GAP\_START\_SELECTIVE\_CONNECTION\_ESTABLISH\_PROC output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

#### Events generated

- [HCI\\_COMMAND\\_STATUS\\_EVENT](#)
- [ACI\\_GAP\\_PROC\\_COMPLETE\\_EVENT](#)
- [HCI\\_LE\\_ENHANCED\\_CONNECTION\\_COMPLETE\\_EVENT](#)

#### 2.4.27 ACI\_GAP\_CREATE\_CONNECTION

Starts the direct connection establishment procedure. An LE\_Create\_Connection call is made to the controller by GAP with the initiator filter policy set to "ignore white list and process connectable advertising packets only for the specified device".

The procedure can be terminated explicitly by the upper layer by issuing the command ACI\_GAP\_TERMINATE\_GAP\_PROC. When this happens, a HCI\_LE\_CREATE\_CONNECTION\_CANCEL call is made to the controller by GAP. On termination of the procedure, a HCI\_LE\_CONNECTION\_COMPLETE\_EVENT (or HCI\_LE\_ENHANCED\_CONNECTION\_COMPLETE\_EVENT if privacy or extended advertising is used) is returned. The procedure can be explicitly terminated by the upper layer by issuing the command ACI\_GAP\_TERMINATE\_GAP\_PROC with the procedure\_code set to 0x40. If controller privacy is enabled and the peer device (advertiser) is in the resolving list, the link layer generates an RPA. If not, the RPA/NRPA generated by the host is used.

**Table 217. ACI\_GAP\_CREATE\_CONNECTION input parameters**

Parameter	Size	Description	Possible values
LE_Scan_Interval	2	<p>Time interval from when the controller started its last LE scan until it begins the subsequent LE scan.</p> <p>Time = N * 0.625 ms</p>	<ul style="list-style-type: none"> <li>• 0x0004 (2.5 ms) ... 0x4000 (10240 ms): legacy advertising</li> </ul>

Parameter	Size	Description	Possible values
			<ul style="list-style-type: none"> <li>• 0x0004 (2.5 ms) ... 0x5DC0 (15000 ms): extended advertising</li> </ul>
LE_Scan_Window	2	<p>Amount of time for the duration of the LE scan. LE_Scan_Window must less than or equal to LE_Scan_Interval. <math>\text{Time} = N * 0.625 \text{ ms}</math></p>	<ul style="list-style-type: none"> <li>• 0x0004 (2.5 ms) ... 0x4000 (10240 ms): legacy advertising</li> <li>• 0x0004 (2.5 ms) ... 0x5DC0 (15000 ms): extended advertising</li> </ul>
Peer_Address_Type	1	The address type of the peer device.	<ul style="list-style-type: none"> <li>• 0x00: Public address</li> <li>• 0x01: Random address</li> </ul>
Peer_Address	6	Public device address or random device address of the device to be connected.	-
Own_Address_Type	1	Own address type: if Privacy is disabled, the address can be public or static random, otherwise, it can be a resolvable private address.	<ul style="list-style-type: none"> <li>• 0x00: Public address</li> <li>• 0x01: Random address</li> <li>• 0x02: Resolvable private address</li> </ul>
Conn_Interval_Min	2	<p>Minimum value for the connection event interval. This shall be less than or equal to Conn_Interval_Max. <math>\text{Time} = N * 1.25 \text{ ms}</math></p>	0x0006 (7.50 ms) ... 0x0C80 (4000 ms)
Conn_Interval_Max	2	<p>Maximum value for the connection event interval. This shall be greater than or equal to Conn_Interval_Min. <math>\text{Time} = N * 1.25 \text{ ms}</math></p>	0x0006 (7.50 ms) ... 0x0C80 (4000 ms)
Conn_Latency	2	Peripheral latency for the connection, in number of connection events.	0x0000 ... 0x01F3
Supervision_Timeout	2	<p>Supervision timeout for the LE link. Must be a multiple of 10 ms, and larger than <math>(1 + \text{connPeripheralLatency}) * \text{connInterval} * 2</math>. <math>\text{Time} = N * 10 \text{ ms}</math></p>	0x000A (100 ms) ... 0x0C80 (32000 ms)
Minimum_CE_Length	2	<p>Information parameter about the minimum length of connection needed for this LE connection. <math>\text{Time} = N * 0.625 \text{ ms}</math></p>	0x0000 (0.000 ms) ... 0xFFFF (40959.375 ms)
Maximum_CE_Length	2	<p>Information parameter about the maximum length of connection needed for this LE connection. <math>\text{Time} = N * 0.625 \text{ ms}</math></p>	0x0000 (0.000 ms) ... 0xFFFF (40959.375 ms)

**Table 218. ACI\_GAP\_CREATE\_CONNECTION output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

#### Events generated

- **HCI\_COMMAND\_STATUS\_EVENT**
- **HCI\_LE\_CONNECTION\_COMPLETE\_EVENT**
- **HCI\_LE\_ENHANCED\_CONNECTION\_COMPLETE\_EVENT**

#### 2.4.28 ACI\_GAP\_TERMINATE\_GAP\_PROC

Terminates the specified GATT procedure. An ACI\_GAP\_PROC\_COMPLETE\_EVENT is returned, with the procedure code set to the corresponding procedure.

**Table 219.** ACI\_GAP\_TERMINATE\_GAP\_PROC input parameters

Parameter	Size	Description	Possible values
Procedure_Code	1	GAP procedure bitmap.	<ul style="list-style-type: none"> <li>• 0x00: No events</li> <li>• 0x01: GAP_LIMITED_DISCOVERY_PROC</li> <li>• 0x02: GAP_GENERAL_DISCOVERY_PROC</li> <li>• 0x04: GAP_NAME_DISCOVERY_PROC</li> <li>• 0x08: GAP_AUTO_CONNECTION_ESTABLISHMENT_PROC</li> <li>• 0x10: GAP_GENERAL_CONNECTION_ESTABLISHMENT_PROC</li> <li>• 0x20: GAP_SELECTIVE_CONNECTION_ESTABLISHMENT_PROC</li> <li>• 0x40: GAP_DIRECT_CONNECTION_ESTABLISHMENT_PROC</li> <li>• 0x80: GAP_OBSERVATION_PROC</li> </ul>

**Table 220.** ACI\_GAP\_TERMINATE\_GAP\_PROC output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-

#### Events generated

- HCI\_COMMAND\_COMPLETE\_EVENT
- ACI\_GAP\_PROC\_COMPLETE\_EVENT

#### 2.4.29 ACI\_GAP\_START\_CONNECTION\_UPDATE

Starts the connection update procedure (only when role is ,central). An HCI\_LE\_CONNECTION\_UPDATE is called. On completion of the procedure, an HCI\_LE\_CONNECTION\_UPDATE\_COMPLETE\_EVENT is returned to the upper layer.

**Table 221.** ACI\_GAP\_START\_CONNECTION\_UPDATE input parameters

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle for which the command is given.	0x0000 ... 0x0EFF
Conn_Interval_Min	2	Minimum value for the connection event interval. This is less than or equal to Conn_Interval_Max. Time = N * 1.25 ms	0x0006 (7.50 ms) ... 0x0C80 (4000.00 ms)
Conn_Interval_Max	2	Maximum value for the connection event interval. This is greater than or equal to Conn_Interval_Min. Time = N * 1.25 ms	0x0006 (7.50 ms) ... 0x0C80 (4000.00 ms)
Conn_Latency	2	Peripheral latency for the connection, in number of connection events.	0x0000 ... 0x01F3
Supervision_Timeout	2	Supervision timeout for the LE link. It is a multiple of 10 ms, and larger than (1 + connPeripheralLatency) * connInterval * 2. Time = N * 10 msec.	0x000A (100 ms) ... 0x0C80 (32000 ms)
Minimum_CE_Length	2	Information parameter about the minimum length of connection needed for this LE connection. Time = N * 0.625 ms	0x0000 (0.000 ms) ... 0xFFFF (40959.375 ms)
Maximum_CE_Length	2	Information parameter about the maximum length of connection needed for this LE connection. Time = N * 0.625 ms	0x0000 (0.000 ms) ... 0xFFFF (40959.375 ms)

**Table 222.** ACI\_GAP\_START\_CONNECTION\_UPDATE output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- `HCI_COMMAND_STATUS_EVENT`
- `HCI_LE_CONNECTION_UPDATE_COMPLETE_EVENT`

**2.4.30****ACI\_GAP\_SEND\_PAIRING\_REQ**

Sends the SM pairing request to start a pairing process. The authentication requirements and IO capabilities must be set before issuing this command, using the `ACI_GAP_SET_IO_CAPABILITY` and `ACI_GAP_SET_AUTHENTICATION_REQUIREMENT` commands. An `ACI_GAP_PAIRING_COMPLETE_EVENT` returned after the pairing process is completed.

**Table 223. ACI\_GAP\_SEND\_PAIRING\_REQ input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle for which the command is given.	0x0000 ... 0x0EFF
Force_Rebond	1	If 1, pairing request is sent even if the device was previously bonded, otherwise pairing request is not sent.	<ul style="list-style-type: none"><li>• 0x00: NO</li><li>• 0x01: YES</li></ul>

**Table 224. ACI\_GAP\_SEND\_PAIRING\_REQ output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- `HCI_COMMAND_STATUS_EVENT`
- `ACI_GAP_PAIRING_COMPLETE_EVENT`

**2.4.31****ACI\_GAP\_RESOLVE\_PRIVATE\_ADDR**

This command tries to resolve the address provided with the IRKs present in its database. If the address is resolved successfully with any one of the IRKs present in the database, it returns success and also the corresponding public/static random address stored with the IRK in the database.

**Table 225. ACI\_GAP\_RESOLVE\_PRIVATE\_ADDR input parameters**

Parameter	Size	Description	Possible values
Address	6	Address to be resolved	-

**Output parameters****Table 226. ACI\_GAP\_RESOLVE\_PRIVATE\_ADDR output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-
Actual_Address	6	The public or static random address of the peer device, distributed during pairing phase.	-

**Events generated**

- `HCI_COMMAND_COMPLETE_EVENT`

**2.4.32****ACI\_GAP\_SET\_BROADCAST\_MODE**

This command, which supports only legacy advertising, puts the device into broadcast mode. A privacy enabled device uses either a resolvable private address, or a non-resolvable private address, as specified in the `Own_Addr_Type` parameter of the command.

**Table 227. ACI\_GAP\_SET\_BROADCAST\_MODE input parameters**

Parameter	Size	Description	Possible values
Advertising_Interval_Min	2	Minimum advertising interval. Time = N * 0.625 ms	0x0020 (20 ms) ... 0x4000 (10240 ms)
Advertising_Interval_Max	2	Maximum advertising interval. Time = N * 0.625 ms	0x0020 (20ms) ... 0x4000 (10240 ms)
Advertising_Type	1	Non connectable advertising type	<ul style="list-style-type: none"> <li>• 0x02: ADV_SCAN_IND (scannable undirected advertising)</li> <li>• 0x03: ADV_NONCONN_IND (non connectable undirected advertising)</li> </ul>
Own_Address_Type	1	If privacy is disabled the address can be public or static random. Otherwise, it can be a resolvable or a non-resolvable private address.	<ul style="list-style-type: none"> <li>• 0x00: Public address</li> <li>• 0x01: Static random address</li> <li>• 0x02: Resolvable private address</li> <li>• 0x03: Non-resolvable private address</li> </ul>
Adv_Data_Length	1	Length of the advertising data in the advertising packet	-
Adv_Data	Adv_Data_Length	Advertising data used by the device while advertising	-
Num_of_Whitelist_Entries	1	Number of devices to add to the white list	-
Peer_Address_Type[i]	1	Address type	<ul style="list-style-type: none"> <li>• 0x00: Public device address</li> <li>• 0x01: Random device address</li> </ul>
Peer_Address[i]	6	Public device address or random device address of the device to be added to the white list	-

**Table 228. ACI\_GAP\_SET\_BROADCAST\_MODE output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

#### Events generated

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

#### 2.4.33 ACI\_GAP\_START\_OBSERVATION\_PROC

Starts an observation procedure, when the device is in observer role. The host enables scanning in the controller. The advertising reports are sent to the upper layer using standard LE advertising report event. If controller privacy is enabled and the peer device (advertiser) is in the resolving list, the link layer generates an RPA, if not, the RPA/NRPA generated by the host is used.

**Table 229. ACI\_GAP\_START\_OBSERVATION\_PROC input parameters**

Parameter	Size	Description	Possible values
LE_Scan_Interval	2	Time interval from when the controller started its last LE scan until it begins the subsequent LE scan.  Time = N * 0.625 ms	<ul style="list-style-type: none"> <li>• 0x0004 (2.5 ms) ... 0x4000 (10240 ms): legacy advertising</li> <li>• 0x0004 (2.5 ms) ... 0x5DC0 (15000 ms): extended advertising</li> </ul>
LE_Scan_Window	2	Amount of time for the duration of the LE scan. LE_Scan_Window is less than or equal to LE_Scan_Interval.  Time = N * 0.625 ms	<ul style="list-style-type: none"> <li>• 0x0004 (2.5 ms) ... 0x4000 (10240 ms): legacy advertising</li> <li>• 0x0004 (2.5 ms) ... 0x5DC0 (15000 ms): extended advertising</li> </ul>
LE_Scan_Type	1	Passive or active scanning. With active scanning SCAN_REQ packets are sent.	<ul style="list-style-type: none"> <li>• 0x00: Passive scanning</li> <li>• 0x01: Active scanning</li> </ul>
Own_Address_Type	1	Own address type: if Privacy is disabled, the address can be public or static random; otherwise, it can be a resolvable private address or a non-resolvable private address.	<ul style="list-style-type: none"> <li>• 0x00: Public device address</li> <li>• 0x01: Random device address</li> <li>• 0x02: Resolvable private address</li> <li>• 0x03: Non resolvable private address</li> </ul>
Filter_Duplicates	1	Enable/disable duplicate filtering.	<ul style="list-style-type: none"> <li>• 0x00: Duplicate filtering disabled</li> <li>• 0x01: Duplicate filtering enabled</li> </ul>
Scanning_Filter_Policy	1	Scanning filter policy: <ul style="list-style-type: none"> <li>• 0x00 Accept all advertisement packets (it is allowed only if controller privacy is enabled). Directed advertising packets which are not addressed for this device is ignored.</li> <li>• 0x01 Ignore advertisement packets from devices not in the white list Only. Directed advertising packets which are not addressed for this device is ignored.</li> <li>• 0x02 Accept all undirected advertisement packets (it is allowed only if controller privacy is enabled). Directed advertisement packets where initiator address is a RPA and directed advertisement packets addressed to this device is accepted.</li> <li>• 0x03 Accept all undirected advertisement packets from devices that are in the white list. Directed advertisement packets where initiator address is RPA and directed advertisement packets addressed to this device is accepted.</li> </ul>	<ul style="list-style-type: none"> <li>• 0x00: Accept all</li> <li>• 0x01: Ignore devices not in the white list</li> <li>• 0x02: Accept all (use resolving list)</li> <li>• 0x03: Ignore devices not in the white list (use resolving list)</li> </ul>

**Table 230. ACI\_GAP\_START\_OBSERVATION\_PROC output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- **HCI\_COMMAND\_STATUS\_EVENT**
- **HCI\_LE\_ADVERTISING\_REPORT\_EVENT**

#### 2.4.34 ACI\_GAP\_GET\_BONDED\_DEVICES

This command gets the list of the devices which are bonded. It returns the number of addresses and the corresponding address types and values.

**Input parameters:** none

**Table 231. ACI\_GAP\_GET\_BONDED\_DEVICES output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-
Num_of_Addresses	1	The number of bonded devices	-
Address_Type[i]	1	Address type.	<ul style="list-style-type: none"><li>• 0x00: Public device address</li><li>• 0x01: Random device address</li></ul>
Address[i]	6	Public device address or random device address of the device to be added to the white list.	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

#### 2.4.35 ACI\_GAP\_IS\_DEVICE\_BONDED

Finds whether the device, whose address is specified in the command, is present in the bonding table. If the device is found the command returns [BLE\\_STATUS\\_SUCCESS](#).

**Table 232. ACI\_GAP\_IS\_DEVICE\_BONDED input parameters**

Parameter	Size	Description	Possible values
Peer_Address_Type	1	Address type.	<ul style="list-style-type: none"><li>• 0x00: Public device address</li><li>• 0x01: Random device address</li></ul>
Peer_Address	6	Address used by the peer device while advertising	-

**Table 233. ACI\_GAP\_IS\_DEVICE\_BONDED output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

#### 2.4.36 ACI\_GAP\_NUMERIC\_COMPARISON\_VALUE\_CONFIRM\_YESNO

This command allows the user to validate/confirm or not the Numeric Comparison value shown through the [ACI\\_GAP\\_Numeric\\_Comparison\\_Value\\_Event](#).

**Table 234. ACI\_GAP\_NUMERIC\_COMPARISON\_VALUE\_CONFIRM\_YESNO input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle for which the command is given.	0x0000 ... 0x0EFF
Confirm_Yes_No	1	0 : The Numeric values showed on both local and peer device are different 1 : The Numeric values showed on both local and peer device are equal!	<ul style="list-style-type: none"><li>• 0x00: No</li><li>• 0x01: Yes</li></ul>

**Table 235. ACI\_GAP\_NUMERIC\_COMPARISON\_VALUE\_CONFIRM\_YESNO output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.4.37**
**ACI\_GAP\_PASSKEY\_INPUT**

This command permits to signal to the stack the input type detected during passkey input.

**Table 236. ACI\_GAP\_PASSKEY\_INPUT input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle for which the command is given.	0x0000 ... 0x0EFF
Input_Type	1	Passkey input type detected	<ul style="list-style-type: none"> <li>• 0x00: Passkey entry started</li> <li>• 0x01: Passkey digit entered</li> <li>• 0x02: Passkey digit erased</li> <li>• 0x03: Passkey cleared</li> <li>• 0x04: Passkey entry completed</li> </ul>

**Table 237. ACI\_GAP\_PASSKEY\_INPUT output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.4.38**
**ACI\_GAP\_GET\_OOB\_DATA**

This command is sent by the server to get (extract) the OOB data generated by the stack.

**Table 238. ACI\_GAP\_GET\_OOB\_DATA input parameters**

Parameter	Size	Description	Possible values
OOB_Data_Type	1	OOB data type	<ul style="list-style-type: none"> <li>• 0x00: TK (legacy pairing)</li> <li>• 0x01: Random (SC)</li> <li>• 0x02: Confirm (SC)</li> </ul>

**Table 239. ACI\_GAP\_GET\_OOB\_DATA output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-
Address_Type	1	Identity address type	<ul style="list-style-type: none"> <li>• 0x00: Public identity address</li> <li>• 0x01: Random (static) identity address</li> </ul>
Address	6	Public or random (static) address of this device	-
OOB_Data_Type	1	OOB data type	<ul style="list-style-type: none"> <li>• 0x00: TK (legacy pairing)</li> <li>• 0x01: Random (SC)</li> <li>• 0x02: Confirm (SC)</li> </ul>
OOB_Data_Len	1	Length of OOB data	16
OOB_Data	16	Local OOB data	-

**Events generated**

- HCI\_COMMAND\_COMPLETE\_EVENT

**2.4.39 ACI\_GAP\_SET\_OOB\_DATA**

This command is sent by the user to input the OOB data received via OOB communication.

**Table 240. ACI\_GAP\_SET\_OOB\_DATA input parameters**

Parameter	Size	Description	Possible values
Device_Type	1	OOB device type	<ul style="list-style-type: none"><li>• 0x00: Local device (Address_Type and Address are not used)</li><li>• 0x01: Remote device</li></ul>
Address_Type	1	Identity address type	<ul style="list-style-type: none"><li>• 0x00: Public identity address</li><li>• 0x01: Random (static) identity address</li></ul>
Address	6	Public or random (static) address of the peer device	-
OOB_Data_Type	1	OOB data type	<ul style="list-style-type: none"><li>• 0x00: TK (legacy pairing)</li><li>• 0x01: Random (SC)</li><li>• 0x02: Confirm (SC)</li></ul>
OOB_Data_Len	1	Length of OOB data	<ul style="list-style-type: none"><li>• 0: SC Random/Confirm generation (OOB_Data and OOB_Data_Type are not used)</li><li>• 1:</li></ul>
OOB_Data	16	Local or remote OOB data received through OOB from peer device (see Device_Type)	-

**Table 241. ACI\_GAP\_SET\_OOB\_DATA output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- HCI\_COMMAND\_COMPLETE\_EVENT

**2.4.40 ACI\_GAP\_ADD\_DEVICES\_TO\_RESOLVING\_LIST**

This command is used to add devices to the list of address translations used to resolve resolvable private addresses in the controller.

**Table 242. ACI\_GAP\_ADD\_DEVICES\_TO\_RESOLVING\_LIST input parameters**

Parameter	Size	Description	Possible values
Num_of_Resolving_list_Entries	1	Number of devices that have to be added to the resolving list.	-
Peer_Identity_Address_Type[i]	1	Identity address type.	<ul style="list-style-type: none"><li>• 0x00: Public identity address</li><li>• 0x01: Random (static) identity address</li></ul>
Peer_Identity_Address[i]	6	Public or random (static) identity address of the peer device	-
Clear_Resolving_List	1	Clear the resolving list	<ul style="list-style-type: none"><li>• 0x00: Do not clear</li><li>• 0x01: Clear before adding</li></ul>

**Table 243. ACI\_GAP\_ADD\_DEVICES\_TO\_RESOLVING\_LIST output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.4.41**
**ACI\_GAP\_REMOVE\_BONDED\_DEVICE**

This command is used to remove a specified device from the bonding table. It returns the number of addresses and the corresponding address types and values.

**Table 244. ACI\_GAP\_REMOVE\_BONDED\_DEVICE input parameters**

Parameter	Size	Description	Possible values
Peer_Identity_Address_Type	1	Identity address type.	<ul style="list-style-type: none"> <li>• 0x00: Public identity address</li> <li>• 0x01: Random (static) identity address</li> </ul>
Peer_Identity_Address	6	Public or random (static) identity address of the peer device	-

**Table 245. ACI\_GAP\_REMOVE\_BONDED\_DEVICE output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.4.42**
**ACI\_GAP\_ADD\_DEVICES\_TO\_LIST**

This command is used by the Host to add specific device addresses to the white and/or resolving lists.

**Table 246. ACI\_GAP\_ADD\_DEVICES\_TO\_LIST input parameters**

Parameter	Size	Description	Possible values
Num_of_Resolving_list_Entries	1	Number of devices to add to the list	-
Address_Type[i]	1	Address type	<ul style="list-style-type: none"> <li>• 0x00: public device address</li> <li>• 0x01:random device address</li> </ul>
Address[i]	6		-
Mode	1		<ul style="list-style-type: none"> <li>• 0x00: append only to the resolving list</li> <li>• 0x01: clear and set only the resolving list</li> <li>• 0x02: append only to the white list</li> <li>• 0x03: clear and set only the white</li> <li>• 0x04: append to both lists</li> <li>• 0x05: clear and set both lists</li> </ul>

**Table 247. ACI\_GAP\_ADD\_DEVICES\_TO\_LIST output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

## 2.4.43 ACI\_GAP\_ADDITIONAL\_BEACON\_START

This command starts an advertising beacon: additional advertising packets can be transmitted independently from the packets transmitted with GAP advertising commands, such as ACI\_GAP\_SET\_DISCOVERABLE or ACI\_GAP\_SET\_LIMITED\_DISCOVERABLE.

**Table 248. ACI\_GAP\_ADDITIONAL\_BEACON\_START input parameters**

Parameter	Size	Description	Possible values
Adv_Interval_Min	2	Minimum advertising interval. Time = N * 0.625 msec.	<ul style="list-style-type: none"> <li>• 0x0020: 20 ms</li> <li>• 0x4000: 10240 ms</li> </ul>
Adv_Interval_Max	2	Maximum advertising interval. Time = N * 0.625 msec.	<ul style="list-style-type: none"> <li>• 0x0020: 20 ms</li> <li>• 0x4000: 10240 ms</li> </ul>
Adv_Channel_Map	1	Advertising channel map. Default: 00000111b (all channels enabled).	Bitmask of <ul style="list-style-type: none"> <li>• 0x01: use Channel 37</li> <li>• 0x02: use Channel 38</li> <li>• 0x04: use Channel 39</li> </ul>
Own_Address_Type	1	Own address type: public or static random	<ul style="list-style-type: none"> <li>• 0x00: Public device address</li> <li>• 0x01: Random device address</li> </ul>
Own_Address	6	Public or random device address	-
PA_Level	1	Power amplifier output level. Output power is indicative and depends on the PCB layout and associated components. Here the values are given at the STM32WB output.	<ul style="list-style-type: none"> <li>• 0x00: -40 dBm</li> <li>• 0x01: -20.85 dBm</li> <li>• 0x02: -19.75 dBm</li> <li>• 0x03: -18.85 dBm</li> <li>• 0x04: -17.6 dBm</li> <li>• 0x05: -16.5 dBm</li> <li>• 0x06: -15.25 dBm</li> <li>• 0x07: -14.1 dBm</li> <li>• 0x08: -13.15 dBm</li> <li>• 0x09: -12.05 dBm</li> <li>• 0x0A: -10.9 dBm</li> <li>• 0x0B: -9.9 dBm</li> <li>• 0x0C: -8.85 dBm</li> <li>• 0x0D: -7.8 dBm</li> <li>• 0x0E: -6.9 dBm</li> <li>• 0x0F: -5.9 dBm</li> <li>• 0x10: -4.95 dBm</li> <li>• 0x11: -4 dBm</li> <li>• 0x12: -3.15 dBm</li> <li>• 0x13: -2.45 dBm</li> <li>• 0x14: -1.8 dBm</li> <li>• 0x15: -1.3 dBm</li> <li>• 0x16: -0.85 dBm</li> <li>• 0x17: -0.5 dBm</li> <li>• 0x18: -0.15 dBm</li> <li>• 0x19: 0 dBm</li> <li>• 0x1A: +1 dBm</li> <li>• 0x1B: +2 dBm</li> <li>• 0x1C: +3 dBm</li> <li>• 0x1D: +4 dBm</li> <li>• 0x1E: +5 dBm</li> <li>• 0x1F: +6 dBm</li> </ul>

**Table 249. ACI\_GAP\_ADDITIONAL\_BEACON\_START output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.4.44****ACI\_GAP\_ADDITIONAL\_BEACON\_STOP**

This command stops the advertising beacon started with ACI\_GAP\_ADDITIONAL\_BEACON\_START.

**Input parameters:** none

**Table 250. ACI\_GAP\_REMOVE\_BONDED\_DEVICE output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.4.45****ACI\_GAP\_ADDITIONAL\_BEACON\_SET\_DATA**

This command sets the data transmitted by the advertising beacon started with ACI\_GAP\_ADDITIONAL\_BEACON\_START. If the advertising beacon is already started, the new data are used in subsequent beacon advertising events.

**Table 251. ACI\_GAP\_ADDITIONAL\_BEACON\_SET\_DATA input parameters**

Parameter	Size	Description	Possible values
Adv_Data_Length	1	Length of Adv_Data in octets	-
Adv_Data	Adv_Data_Length	Advertising data used by the device while advertising	-

**Table 252. ACI\_GAP\_ADDITIONAL\_BEACON\_SET\_DATA output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.4.46****ACI\_GAP\_ADV\_SET\_CONFIGURATION**

This command is used to set the extended advertising configuration for one advertising set. In association with ACI\_GAP\_ADV\_SET\_SCAN RESP DATA, ACI\_GAP\_ADV\_SET\_ADV DATA, and ACI\_GAP\_ADV\_SET\_ENABLE, it enables to start extended advertising.

These commands must be used to replace ACI\_GAP\_SET\_DISCOVERABLE, ACI\_GAP\_SET\_LIMITED\_DISCOVERABLE, ACI\_GAP\_SET\_DIRECT\_CONNECTABLE, ACI\_GAP\_SET\_NON\_CONNECTABLE, ACI\_GAP\_SET\_UNDIRECTED\_CONNECTABLE, and ACI\_GAP\_SET\_BROADCAST\_MODE, supporting only legacy advertising.

If bit 0 of Adv\_Mode is set, the Own\_Address\_Type parameter is ignored and Own\_Address is set with the ACI\_GAP\_ADV\_SET\_RANDOM\_ADDRESS command. This mode is valid only for non-connectable advertising.

If bit 1 of Adv\_Mode is set, the primary advertisement PHY is set to LE coded (not supported on STM32WB), otherwise, the default primary advertisement PHY is LE 1M.

To configure a periodic advertising set, use the command ACI\_GAP\_ADV\_SET\_PERIODIC\_PARAMETERS (in association with ACI\_GAP\_ADV\_SET\_PERIODIC\_DATA and ACI\_GAP\_ADV\_SET\_PERIODIC\_ENABLE) instead of this one (feature not supported on STM32WB).

**Table 253. ACI\_GAP\_ADV\_SET\_CONFIGURATION input parameters**

Parameter	Size	Description	Possible values
Adv_Mode	1	Bitmap of extended advertising modes	<p>Bitmask of:</p> <ul style="list-style-type: none"> <li>• 0x01: use specific random address</li> <li>• 0x02: use LE coded as primary advertising PHY (not supported by STM32WB devices)</li> </ul>
Advertising_Handle	1	Used to identify an advertising set	0x00 ... 0xEF
Adv_Event_Properties	2	Type of advertising event	<p>Bitmask of:</p> <ul style="list-style-type: none"> <li>• 0x0001: connectable advertising</li> <li>• 0x0002: scannable advertising</li> <li>• 0x0004: directed advertising</li> <li>• 0x0008: high duty cycle directed connectable advertising</li> <li>• 0x0010: use legacy advertising PDUs</li> <li>• 0x0020: anonymous advertising</li> <li>• 0x0040: include TxPower in at least one advertising PDU</li> </ul>
Primary_Adv_Interval_Min	4	Minimum advertising interval. Time = N * 0.625 ms.	0x00000020 (20.0 ms) ... 0x00FFFFFF (10485759.375 ms)
Primary_Adv_Interval_Max	4	Maximum advertising interval. Time = N * 0.625 ms.	0x00000020 (20.0 ms) ... 0x00FFFFFF (10485759.375 ms)
Primary_Adv_Channel_Map	1	Advertising channel map.	<p>Bitmask of:</p> <ul style="list-style-type: none"> <li>• 0x01: use channel 37</li> <li>• 0x02: use channel 38</li> <li>• 0x04: use channel 39</li> </ul>
Own_Address_Type	1	Own address type: if Privacy is disabled, it can be public or static random, otherwise, it can be a resolvable or a non-resolvable private address.	<ul style="list-style-type: none"> <li>• 0x00: public address</li> <li>• 0x01: static random address</li> <li>• 0x02: resolvable private address</li> <li>• 0x03: non-resolvable private address</li> </ul>
Peer_Address_Type	1	Address type of the peer device	<ul style="list-style-type: none"> <li>• 0x00: public device or public identity address</li> <li>• 0x01: static random address</li> <li>• 0x02: random device or random (static) identity address</li> </ul>
Peer_Address	6	Public or random device, public or random (static) identity address of the device to connect	-
Adv_Filter_Policy	1	Advertising filter policy	<ul style="list-style-type: none"> <li>• 0x00: process scan and connection requests from all devices (white list not in use)</li> <li>• 0x01: process connection requests from all devices and scan requests only from devices in the white list</li> <li>• 0x02: process scan requests from all devices and connection requests only from devices in the white list</li> <li>• 0x03: process scan and connection requests only from devices in the white list</li> </ul>
Adv_TX_Power	1	Advertising TX power. Units: dBm.	<ul style="list-style-type: none"> <li>• 127: host has no preferences</li> <li>• -127 ... 20</li> </ul>
Secondary_Adv_Max_Skip	1	Secondary advertising maximum skip	<ul style="list-style-type: none"> <li>• 0x00: AUX_ADV_IND must be sent prior to the next advertising event</li> <li>• 0x01 ... 0xFF: maximum advertising events that the controller can skip before sending the AUX_ADV_IND packets on the secondary advertising physical channel</li> </ul>
Secondary_Adv_PHY	1	Secondary advertising PHY	<ul style="list-style-type: none"> <li>• 0x01: secondary advertisement PHY is LE 1M</li> <li>• 0x02: secondary advertisement PHY is LE 2M</li> <li>• 0x03: secondary advertisement PHY is LE coded (not supported by STM32WB devices)</li> </ul>
Adv_SID	1	Value of the advertising SID subfield in the ADI field of PDU	0x00 ... 0x0F

Parameter	Size	Description	Possible values
Scan_Req_Notification_Enable	1	Scan request notifications	<ul style="list-style-type: none"> <li>• 0x00: scan disabled</li> <li>• 0x01: scan enabled</li> </ul>

**Table 254. ACI\_GAP\_ADV\_SET\_CONFIGURATION output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

#### Events generated

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

### 2.4.47 ACI\_GAP\_ADV\_SET\_ENABLE

This command is used to request the controller to enable/disable one or more extended advertising sets.

**Table 255. ACI\_GAP\_ADV\_SET\_ENABLE input parameters**

Parameter	Size	Description	Possible values
Enable	1	Enable/disable advertising	<ul style="list-style-type: none"> <li>• 0x00: advertising disabled</li> <li>• 0x01: advertising enabled</li> </ul>
Num_Sets	1	Number of advertising sets	<ul style="list-style-type: none"> <li>• 0x00: disable all advertising sets</li> <li>• 0x01 ... 0x3F: number of advertising sets to enable/disable</li> </ul>
Advertising_Handle[i]	1	Used to identify an advertising set	0x00 ... 0xEF
Duration[i]	2	Duration of advertising set. Time = N * 10 ms.	<ul style="list-style-type: none"> <li>• 0x0000 (0 ms): no advertising duration</li> <li>• 0x0001 (10 ms) ... 0xFFFF (655350 ms): advertising duration</li> </ul>
Max_Extended_Advertising_Events[i]	1	Maximum number of advertising events	<ul style="list-style-type: none"> <li>• 0x00: no maximum number of advertising events</li> <li>• 0x01 ... 0xFF: maximum number of extended advertising events the controller attempts to send prior to terminate the extended advertising</li> </ul>

**Table 256. ACI\_GAP\_ADV\_SET\_ENABLE output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

#### Events generated

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

### 2.4.48 ACI\_GAP\_ADV\_SET\_ADV\_DATA

This command is used to request the controller to enable/disable one or more extended advertising sets.

**Table 257. ACI\_GAP\_ADV\_SET\_ADV\_DATA input parameters**

Parameter	Size	Description	Possible values
Advertising_Handle	1	Used to identify an advertising set	0x00 ... 0xEF

Parameter	Size	Description	Possible values
Operation	1	Advertising operation	<ul style="list-style-type: none"> <li>• 0x00: Intermediate fragment of fragmented extended advertising data</li> <li>• 0x01: first fragment of fragmented extended advertising data</li> <li>• 0x02: last fragment of fragmented extended advertising data</li> <li>• 0x03: complete extended advertising data</li> <li>• 0x04: unchanged data (just update the advertising DID)</li> </ul>
Fragment_Preference	1	Fragment preference	<ul style="list-style-type: none"> <li>• 0x00: the controller can fragment data</li> <li>• 0x01: the controller must not fragment data, or minimize fragmentation</li> </ul>
Advertising_Data_Length	1	Length of Advertising_Data in octets	-
Advertising_Data	Advertising_Data_Length	Data formatted as defined in Bluetooth spec. v.5.2 [Vol 3, Part C, 11].	-

**Table 258. ACI\_GAP\_ADV\_SET\_ADV\_DATA output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

#### Events generated

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

#### 2.4.49 ACI\_GAP\_ADV\_SET\_SCAN\_RESP\_DATA

This command is used to provide scan response data used during extended advertising.

**Table 259. ACI\_GAP\_ADV\_SET\_SCAN\_RESP\_DATA input parameters**

Parameter	Size	Description	Possible values
Advertising_Handle	1	Used to identify an advertising set	0x00 ... 0xEF
Operation	1	Advertising operation	<ul style="list-style-type: none"> <li>• 0x00: intermediate fragment of fragmented extended advertising data</li> <li>• 0x01: first fragment of fragmented extended advertising data</li> <li>• 0x02: last fragment of fragmented extended advertising data</li> <li>• 0x03: complete scan response data</li> </ul>
Fragment_Preference	1	Fragment preference	<ul style="list-style-type: none"> <li>• 0x00: the controller may fragment all data</li> <li>• 0x01: the controller must not fragment data, or minimize fragmentation</li> </ul>

Parameter	Size	Description	Possible values
Scan_Response_Data_Length	1	Length of Scan_Response_Data in octets	-
Scan_Response_Data	Scan_Response_Data_Length	Data formatted as defined in Bluetooth spec. v.5.2 [Vol 3, Part C, 11].	-

**Table 260. ACI\_GAP\_ADV\_SET\_SCAN\_RESP\_DATA output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.4.50 ACI\_GAP\_ADV\_REMOVE\_SET**

This command is used to request the controller to enable/disable one or more extended advertising sets.

**Table 261. ACI\_GAP\_ADV\_REMOVE\_SET input parameters**

Parameter	Size	Description	Possible values
Advertising_Handle	1	Used to identify an advertising set	0x00 ... 0xEF

**Table 262. ACI\_GAP\_ADV\_REMOVE\_SET output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.4.51 ACI\_GAP\_ADV\_CLEAR\_SETS**

This command is used to remove all existing advertising sets from the controller.

**Input parameters:** none

**Table 263. ACI\_GAP\_ADV\_CLEAR\_SETS output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.4.52 ACI\_GAP\_ADV\_SET\_RANDOM\_ADDRESS**

This command is used to set the random device address of an advertising set configured to use specific random address.

**Table 264. ACI\_GAP\_ADV\_SET\_RANDOM\_ADDRESS input parameters**

Parameter	Size	Description	Possible values
Advertising_Handle	1	Used to identify an advertising set	0x00 ... 0xEF
Random_Address	6	Random device address	-

**Table 265. ACI\_GAP\_ADV\_SET\_RANDOM\_ADDRESS output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

#### 2.4.53 ACI\_GAP\_EXT\_START\_SCAN

This command is used to start a scanning procedure when the extended advertising feature is supported. It can be used instead of ACI\_GAP\_START\_LIMITED\_DISCOVERY\_PROC, ACI\_GAP\_START\_GENERAL\_DISCOVERY\_PROC, ACI\_GAP\_START\_GENERAL\_CONNECTION\_ESTABLISH\_PROC, ACI\_GAP\_START\_SELECTIVE\_CONNECTION\_ESTABLISH\_PROC, or ACI\_GAP\_START\_OBSERVATION\_PROC, as it is more generic.

The Scanning\_PHYs and subsequent parameters are used to specify the scanning parameters as defined for HCI\_LE\_SET\_EXTENDED\_SCAN\_PARAMETERS. For more details, refer to Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.64].

**Table 266. ACI\_GAP\_EXT\_START\_SCAN input parameters**

Parameter	Size	Description	Possible values
Scan Mode	1	Not used, set it to 0	-
Procedure	1	Scan procedure	<ul style="list-style-type: none"> <li>• 0x01: GAP_LIMITED_DISCOVERY_PROC</li> <li>• 0x02: GAP_GENERAL_DISCOVERY_PROC</li> <li>• 0x10: GAP_GENERAL_CONNECTION_ESTABLISHMENT_PROC</li> <li>• 0x20: GAP_SELECTIVE_CONNECTION_ESTABLISHMENT_PROC</li> <li>• 0x80: GAP_OBSERVATION_PROC</li> </ul>
Own_Address_Type	1	Own address type: if Privacy is disabled, the address can be public or static random; otherwise, it can be a resolvable or a non-resolvable private address.	<ul style="list-style-type: none"> <li>• 0x00: Public address</li> <li>• 0x01: Static random address</li> <li>• 0x02: Resolvable private address</li> <li>• 0x03: Non-resolvable private address</li> </ul>
Filter_Duplicate	1	Enable/disable duplicate filtering	<ul style="list-style-type: none"> <li>• 0x00: Duplicate filtering disabled</li> <li>• 0x01: Duplicate filtering enabled</li> </ul>
Duration	2	Scan duration. Time = N * 10 ms.	<ul style="list-style-type: none"> <li>• 0x000 (0 ms) : Scan continuously until explicitly disabled</li> <li>• 0x001 (10 ms) ... 0xFFFF (655350 ms): Scan duration</li> </ul>
Period	2	Scan period. Time = N * 1.28 s.	<ul style="list-style-type: none"> <li>• 0x0000 (0 ms): Scan continuously</li> <li>• 0x0001 (1280 ms) ... 0xFFFF (83884800 ms): Time interval from when the controller started its last Scan_Duration until it begins the subsequent one</li> </ul>

Parameter	Size	Description	Possible values
Scanning_Filter_Policy	1	<p>Determines how the scanner's Link Layer processes advertising and scan response PDUs.</p> <p>There are two primary filter policies:</p> <ul style="list-style-type: none"> <li>• Unfiltered: the Link Layer processes all advertising and scan response PDUs (the Filter Accept List is not used)</li> <li>• Filtered: the Link Layer processes advertising and scan response PDUs only from devices in the Filter Accept List.</li> </ul> <p>With extended scanning filter policies, a directed advertising PDU accepted by the primary filter policy must be ignored unless the TargetA field is identical to the scanner's device address, or TargetA field is a resolvable private address.</p>	<ul style="list-style-type: none"> <li>• 0x00: Basic unfiltered scanning filter policy</li> <li>• 0x01: Basic filtered scanning filter policy</li> <li>• 0x02: Extended unfiltered scanning filter policy</li> <li>• 0x03: Extended filtered scanning filter policy</li> </ul>
Scanning_PHYs	1	Scan PHYs	<p>Bitmask of:</p> <ul style="list-style-type: none"> <li>• 0x01: Scan advertisements on LE 1M PHY</li> <li>• 0x04: Scan advertisements on LE coded PHY (not supported by STM32WB)</li> </ul>
Scan_Type[i]	1	Passive or active scanning. With passive scanning, no scan request PDUs are sent	<ul style="list-style-type: none"> <li>• 0x00: Passive scanning</li> <li>• 0x01: Active scanning</li> </ul>
Scan_Interval[i]	1	Time interval from when the controller started its last scan until it begins the subsequent one on the primary advertising physical channel. Time = N * 0.625 ms.	<ul style="list-style-type: none"> <li>• 0x0004 (2.5 ms) ... 0x5DC0 (15000 ms): STM32WB</li> <li>• 0x0004 (2.5 ms) ... 0xFFFF (40959.375 ms): STM32WBA</li> </ul>
Scan_Window[i]	2	Scan duration on the primary advertising physical channel. Time = N * 0.625 ms.	<ul style="list-style-type: none"> <li>• 0x0004 (2.5 ms) ... 0x5DC0 (15000 ms): STM32WB</li> <li>• 0x0004 (2.5 ms) ... 0xFFFF (40959.375 ms): STM32WBA</li> </ul>

**Table 267. ACI\_GAP\_EXT\_START\_SCAN output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- HCI\_COMMAND\_STATUS\_EVENT
- HCI\_LE\_EXTENDED\_ADVERTISING\_REPORT\_EVENT
- ACI\_GAP\_PROC\_COMPLETE\_EVENT

**2.4.54****ACI\_GAP\_EXT\_CREATE\_CONNECTION**

This command is used to create a connection with the local device in the Central role to an advertiser when the extended advertising feature is supported. This command can be used instead of ACI\_GAP\_CREATE\_CONNECTION or ACI\_GAP\_START\_AUTO\_CONNECTION\_ESTABLISH\_PROC, as it is more generic.

The Advertising\_Handle and Subevent parameters must be used for a connection to a periodic advertiser, otherwise set them to 0xFF. These parameters are ignored by STM32WB devices.

The Initiating\_PHYSs and subsequent parameters are used to specify the initiating parameters as defined for HCI\_LE\_EXTENDED\_CREATE\_CONNECTION: for more details, refer to Bluetooth spec. v.5.4 [Vol 4, Part E, 7.8.66].

**Table 268. ACI\_GAP\_EXT\_CREATE\_CONNECTION input parameters**

Parameter	Size	Description	Possible values
Initiating_Mode	1	Not used, set it to 0	-
Procedure	1	Connection procedure	<ul style="list-style-type: none"> <li>• 0x08: GAP_AUTO_CONNECTION_ESTABLISHMENT_PROC</li> <li>• 0x40: GAP_DIRECT_CONNECTION_ESTABLISHMENT_PROC</li> </ul>
Own_Address_Type	1	Own address type: if privacy is disabled, the address can be public or static random, otherwise, it can be a resolvable private address.	<ul style="list-style-type: none"> <li>• 0x00: Public address</li> <li>• 0x01: Static random address</li> <li>• 0x02: Resolvable private address</li> </ul>
Peer_Address_Type	1	Address type of the peer device	<ul style="list-style-type: none"> <li>• 0x00: Public address</li> <li>• 0x01: Random address</li> </ul>
Peer_Address	6	Public or random device, public or random (static) identity address of the device to connect	-
Advertising_Handle	1	Used to identify the subevent where a connection request must be initiated from a periodic advertising train	<ul style="list-style-type: none"> <li>• 0xFF: parameter not used</li> <li>• 0x00 ... 0xEF</li> </ul>
Subevent	1	Subevent where the connection request must be sent	<ul style="list-style-type: none"> <li>• 0xFF: parameter not used</li> <li>• 0x00 ... 0xEF</li> </ul>
Initiator_Filter_Policy	1	Initiator filter policy	<ul style="list-style-type: none"> <li>• 0x00: Filter accept list is not used to determine which advertiser to connect to</li> <li>• 0x01: Filter accept list is used to determine which advertiser to connect to (Peer_Address_Type and Peer_Address are ignored)</li> </ul>
Initiating_PHYSs	1	Initiating_PHYSs	Bitmask of: <ul style="list-style-type: none"> <li>• 0x01: Scan connectable advertisements on the LE 1M PHY</li> <li>• 0x02: Scan connectable advertisements on the LE 2M PHY</li> </ul>

Parameter	Size	Description	Possible values
			<ul style="list-style-type: none"> <li>• 0x04: Scan connectable advertisements on the LE coded PHY (not supported by STM32WB)</li> </ul>
Scan_Interval[i]	2	Time interval from when the controller started its last scan until it begins the subsequent one on the primary advertising physical channel. Time = N * 0.625 ms.	<ul style="list-style-type: none"> <li>• 0x0004 (2.5 ms) ... 0x5DC0 (15000 ms): STM32WB</li> <li>• 0x0004 (2.5 ms) ... 0xFFFF (40959.375 ms): STM32WBA</li> </ul>
Scan_Window[i]	2	Duration of the scan on the primary advertising physical channel. Time = N * 0.625 ms.	<ul style="list-style-type: none"> <li>• 0x0004 (2.5 ms) ... 0x5DC0 (15000 ms): STM32WB</li> <li>• 0x0004 (2.5 ms) ... 0xFFFF (40959.375 ms): STM32WBA</li> </ul>
Conn_Interval_Min[i]	2	Minimum value for the connection event interval. Time = N * 1.25 ms.	0x0006 (7.5 ms) ... 0x0C80 (4000 ms)
Conn_Interval_Max[i]	2	Minimum value for the connection event interval. Time = N * 1.25 ms.	0x0006 (7.5 ms) ... 0x0C80 (4000 ms)
Conn_Latency[i]	2	Maximum peripheral latency for the connection in number of connection events.	0x0000 ... 0x01F3
Supervision_Timeout[i]	2	Supervision timeout for the LE link. Must be a multiple of 10 ms and larger than (1 + connPeripheralLatency) * connInterval * 2. Time = N * 10 ms	0x000A (100 ms) ... 0x0C80 (32000 ms)
Min_CE_Length[i]	2	Information parameter about the minimum length for this LE connection. Time = N * 0.625 ms.	0x0000 (0.000 ms) ... 0xFFFF (40959.375 ms)
Max_CE_Length[i]	2	Information parameter about the maximum length for this LE connection. Time = N * 0.625 ms.	0x0000 (0.000 ms) ... 0xFFFF (40959.375 ms)

**Table 269. ACI\_GAP\_EXT\_CREATE\_CONNECTION output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- **HCI\_COMMAND\_STATUS\_EVENT**
- **HCI\_LE\_EXTENDED\_ADVERTISING\_REPORT\_EVENT**
- **ACI\_GAP\_PROC\_COMPLETE\_EVENT**

## 2.5

**GATT/ATT commands**

In Table 270 "Y" means that the corresponding command applies to the dedicated BLE stack variant, namely LO / BO / PO / BF / LB, respectively, for Link layer only / Beacon only / Peripheral only / Basic feature / Link layer only basic.

**Table 270. GATT/ATT commands list**

Command	OpCode	LO	PO	BO	BF	LB
ACI_GATT_INIT	0xFD01	-	Y	-	Y	-
ACI_GATT_ADD_SERVICE	0xFD02	-	Y	-	Y	-
ACI_GATT_INCLUDE_SERVICE	0xFD03	-	Y	-	Y	-
ACI_GATT_ADD_CHAR	0xFD04	-	Y	-	Y	-
ACI_GATT_ADD_CHAR_DESC	0xFD05	-	Y	-	Y	-
ACI_GATT_UPDATE_CHAR_VALUE	0xFD06	-	Y	-	Y	-
ACI_GATT_DEL_CHAR	0xFD07	-	Y	-	Y	-
ACI_GATT_DEL_SERVICE	0xFD08	-	Y	-	Y	-
ACI_GATT_DEL_INCLUDE_SERVICE	0xFD09	-	Y	-	Y	-
ACI_GATT_SET_EVENT_MASK	0xFD0A	-	Y	-	Y	-
ACI_GATT_EXCHANGE_CONFIG	0xFD0B	-	-	-	Y	-
ACI_ATT_FIND_INFO_REQ	0xFD0C	-	-	-	Y	-
ACI_ATT_FIND_BY_TYPE_VALUE_REQ	0xFD0D	-	-	-	Y	-
ACI_ATT_READ_BY_TYPE_REQ	0xFD0E	-	-	-	Y	-
ACI_ATT_READ_BY_GROUP_TYPE_REQ	0xFD0F	-	-	-	Y	-
ACI_ATT_PREPARE_WRITE_REQ	0xFD10	-	-	-	Y	-
ACI_ATT_EXECUTE_WRITE_REQ	0xFD11	-	-	-	Y	-
ACI_GATT_DISC_ALL_PRIMARY_SERVICES	0xFD12	-	-	-	Y	-
ACI_GATT_DISC_PRIMARY_SERVICE_BY_UUID	0xFD13	-	-	-	Y	-
ACI_GATT_FIND_INCLUDED_SERVICES	0xFD14	-	-	-	Y	-
ACI_GATT_DISC_ALL_CHAR_OF_SERVICE	0xFD15	-	-	-	Y	-
ACI_GATT_DISC_CHAR_BY_UUID	0xFD16	-	-	-	Y	-
ACI_GATT_DISC_ALL_CHAR_DESC	0xFD17	-	-	-	Y	-
ACI_GATT_READ_CHAR_VALUE	0xFD18	-	-	-	Y	-
ACI_GATT_READ_USING_CHAR_UUID	0xFD19	-	-	-	Y	-
ACI_GATT_READ_LONG_CHAR_VALUE	0xFD1A	-	-	-	Y	-
ACI_GATT_READ_MULTIPLE_CHAR_VALUE	0xFD1B	-	-	-	Y	-
ACI_GATT_WRITE_CHAR_VALUE	0xFD1C	-	-	-	Y	-
ACI_GATT_WRITE_LONG_CHAR_VALUE	0xFD1D	-	-	-	Y	-
ACI_GATT_WRITE_CHAR_RELIABLE	0xFD1E	-	-	-	Y	-
ACI_GATT_WRITE_LONG_CHAR_DESC	0xFD1F	-	-	-	Y	-
ACI_GATT_READ_LONG_CHAR_DESC	0xFD20	-	-	-	Y	-
ACI_GATT_WRITE_CHAR_DESC	0xFD21	-	-	-	Y	-
ACI_GATT_READ_CHAR_DESC	0xFD22	-	-	-	Y	-
ACI_GATT_WRITE_WITHOUT_RESP	0xFD23	-	-	-	Y	-
ACI_GATT_SIGNED_WRITE_WITHOUT_RESP	0xFD24	-	-	-	Y	-

Command	OpCode	LO	PO	BO	BF	LB
ACI_GATT_CONFIRM_INDICATION	0xFD25	-	-	-	Y	-
ACI_GATT_WRITE_RESP	0xFD26	-	Y	-	Y	-
ACI_GATT_ALLOW_READ	0xFD27	-	Y	-	Y	-
ACI_GATT_SET_SECURITY_PERMISSION	0xFD28	-	Y	-	Y	-
ACI_GATT_SET_DESC_VALUE	0xFD29	-	Y	-	Y	-
ACI_GATT_READ_HANDLE_VALUE	0xFD2A	-	Y	-	Y	-
ACI_GATT_UPDATE_CHAR_VALUE_EXT	0xFD2C	-	Y	-	Y	-
ACI_GATT_DENY_READ	0xFD2D	-	Y	-	Y	-
ACI_GATT_SET_ACCESS_PERMISSION	0xFD2E	-	Y	-	Y	-
ACI_GATT_STORE_DB	0xFD30	-	Y	-	Y	-
ACI_GATT_SET_MULT_NOTIFICATION	0xFD31	-	-	-	-	-
ACI_GATT_READ_MULTIPLE_VAR_CHAR_VALUE	0xFD32	-	-	-	-	-

## 2.5.1 ACI\_GATT\_INIT

Initializes the GATT layer for server and client roles, and adds the GATT service with service changed characteristic. Until this command is issued, the GATT channel does not process any command, even if the connection is opened. This command must be given before using any of the GAP features.

**Input parameters:** none

**Table 271. ACI\_GATT\_INIT output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

### Events generated

- HCI\_COMMAND\_COMPLETE\_EVENT

## 2.5.2 ACI\_GATT\_ADD\_SERVICE

### Description

Adds a service to GATT server. When a service is created in the server, the host needs to reserve the handle ranges for this service using the Max\_Attribute\_Records parameter, which specifies the maximum number of attribute records that can be added to this service (including the service attribute, include attribute, characteristic attribute, characteristic value attribute, and characteristic descriptor attribute). Handle of the created service is returned in command complete event. Service declaration is taken from the service pool. The attributes for characteristics and descriptors are allocated from the attribute pool.

**Table 272. ACI\_GATT\_ADD\_SERVICE input parameters**

Parameter	Size	Description	Possible values
Service_UUID_Type	1	UUID type (the selected value modifies the parameter Service_UUID)	<ul style="list-style-type: none"><li>0x01: 16-bit UUID</li><li>0x02: 128-bit UUID</li></ul>
Service_UUID_16 or Service_UUID_128	2 or 16	16- or 128-bit UUID	-
Service_Type	1	Service type	<ul style="list-style-type: none"><li>0x01: Primary service</li><li>0x02: Secondary service</li></ul>
Max_Attribute_Records	1	Maximum number of attribute records that can be added to this service	-

**Table 273. ACI\_GATT\_ADD\_SERVICE output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-
Service_Handle	2	Handle of the Service. When this service is added, a handle is allocated by the server for this service. Server also allocates a range of handles for this service from serviceHandle to <serviceHandle + max_attr_records - 1>	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

### 2.5.3 ACI\_GATT\_INCLUDE\_SERVICE

Includes a service given by Include\_Start\_Handle and Include\_End\_Handle to another service given by Service\_Handle. Attribute server creates an "include" definition attribute and returns the handle of this attribute in Included\_Handle.

**Table 274. ACI\_GATT\_INCLUDE\_SERVICE input parameters**

Parameter	Size	Description	Possible values
Service_Handle	2	Handle of the service to whom another service must be included	-
Include_Start_Handle	2	Start handle of the service that must be included in service	-
Include_End_Handle	2	End handle of the service that must be included in service	-
Include_UUID_Type	1	UUID type: 0x01 = 16 bits UUID while 0x02 = 128 bits UUID	-
Include_UUID	2 or 16	16- or 128-bit UUID	-

**Table 275. ACI\_GATT\_INCLUDE\_SERVICE output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-
Include_Handle	2	Handle of the included declaration	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

### 2.5.4 ACI\_GATT\_ADD\_CHAR

This command adds a characteristic to a service, and returns the handle of the declaration attribute. The attribute that holds the characteristic value is always located at the next handle (Char\_Handle + 1). The characteristic value is immediately followed, in order, by:

- the server characteristic configuration descriptor if CHAR\_PROP\_BROADCAST is selected
- the client characteristic configuration descriptor if CHAR\_PROP\_NOTIFY or CHAR\_PROP\_INDICATE is selected
- the characteristic extended properties descriptor if CHAR\_PROP\_EXT is selected

As an example, if CHAR\_PROP\_NOTIFY is selected, and CHAR\_PROP\_BROADCAST and CHAR\_PROP\_EXT are not, then the client characteristic configuration attribute handle is Char\_Handle + 2.

Other descriptors can be added to the characteristic by calling the ACI\_GATT\_ADD\_CHAR\_DESC command immediately after this one.

**Table 276. ACI\_GATT\_ADD\_CHAR input parameters**

Parameter	Size	Description	Possible values
Service_Handle	2	Handle of the service to whom the characteristic is added	-
Char_UUID_Type	1	UUID type	<ul style="list-style-type: none"> <li>• 0x01 = 16-bit UUID</li> <li>• 0x02 = 128-bit UUID</li> </ul>
Char_UUID	2 or 16	16- or 128-bit UUID	-
Char_Value_Length	2	Maximum length of the characteristic value	-
Char_Properties	1	Characteristic properties	<p>Bitmask of:</p> <ul style="list-style-type: none"> <li>• 0x00: CHAR_PROP_NONE</li> <li>• 0x01: CHAR_PROP_BROADCAST (broadcast)</li> <li>• 0x02: CHAR_PROP_READ (read)</li> <li>• 0x04: CHAR_PROP_WRITE_WITHOUT_RESP (write without resp)</li> <li>• 0x08: CHAR_PROP_WRITE (write)</li> <li>• 0x10: CHAR_PROP_NOTIFY (notify)</li> <li>• 0x20: CHAR_PROP_INDICATE (indicate)</li> <li>• 0x40: CHAR_PROP_SIGNED_WRITE (authenticated signed writes)</li> <li>• 0x80: CHAR_PROP_EXT (extended properties)</li> </ul>
Security_Permissions	1	Security permission flags	<p>Bitmask of:</p> <ul style="list-style-type: none"> <li>• 0x00: None</li> <li>• 0x01: AUTHEN_READ (need authentication to read)</li> <li>• 0x02: AUTHOR_READ (need authorization to read)</li> <li>• 0x04: ENCRY_READ (need encryption to read)</li> <li>• 0x08: AUTHEN_WRITE (need authentication to write)</li> <li>• 0x10: AUTHOR_WRITE (need authorization to write)</li> <li>• 0x20: ENCRY_WRITE (need encryption to write)</li> </ul>
GATT_Evt_Mask	1	GATT event mask	<p>Bitmask of:</p> <ul style="list-style-type: none"> <li>• 0x00: GATT_DONT_NOTIFY_EVENTS</li> <li>• 0x01: GATT_NOTIFY_ATTRIBUTE_WRITE</li> <li>• 0x02: GATT_NOTIFY_WRITE_REQ_AND_WAIT_FOR_APPL_RESP</li> <li>• 0x04: GATT_NOTIFY_READ_REQ_AND_WAIT_FOR_APPL_RESP</li> <li>• 0x08: GATT_NOTIFY_NOTIFICATION_COMPLETION</li> </ul>
Enc_Key_Size	1	Minimum encryption key size required to read the characteristic	0x07 ... 0x10
Is_Variable	1	Specifies if the characteristic value has a fixed or a variable length	<ul style="list-style-type: none"> <li>• 0x00: Fixed length</li> <li>• 0x01: Variable length</li> </ul>

**Table 277. ACI\_GATT\_ADD\_CHAR output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-
Char_Handle	2	Handle of the added characteristic. It is the handle of the characteristic declaration. The attribute that holds the characteristic value is allocated at the next handle, followed by the client characteristic configuration descriptor if the characteristic has CHAR_PROP_NOTIFY or CHAR_PROP_INDICATE properties.	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.5.5****ACI\_GATT\_ADD\_CHAR\_DESC**

Adds a characteristic descriptor to a service. This command allocates the new handle for the descriptor after the current one. It is advisable to call it following the call of ACI\_GATT\_ADD\_CHAR that created the characteristic containing this descriptor.

**Table 278. ACI\_GATT\_ADD\_CHAR\_DESC input parameters**

Parameter	Size	Description	Possible values
Service_Handle	2	Handle of service to whom the characteristic belongs	-
Char_Handle	2	Handle of the characteristic to whom description must be added	-
Char_Desc_Uuid_Type	1	UUID type	<ul style="list-style-type: none"> <li>• 0x01 = 16 bits</li> <li>• 0x02 = 128 bits</li> </ul>
Char_Desc_Uuid	2 or 16	16- or 128-bit UUID	-
Char_Desc_Value_Max_Length	1	Maximum length of the descriptor value	-
Char_Desc_Value_Length	1	Current length of the characteristic description value	-
Char_Desc_Value	Char_Desc_Value_Length	Value of the characteristic description	-
Security_Permissions	1	Security permission flags	<p>Bitmask of:</p> <ul style="list-style-type: none"> <li>• 0x00: None</li> <li>• 0x01: AUTHEN_READ (need authentication to read)</li> <li>• 0x02: AUTHOR_READ (need authorization to read)</li> <li>• 0x04: ENCRY_READ (need encryption to read)</li> <li>• 0x08: AUTHEN_WRITE (need authentication to write)</li> <li>• 0x10: AUTHOR_WRITE (need authorization to write)</li> <li>• 0x20: ENCRY_WRITE (need encryption to write)</li> </ul>

Parameter	Size	Description	Possible values
Access_Permissions	1	Access permission	<p>Bitmask of:</p> <ul style="list-style-type: none"> <li>• 0x00: None</li> <li>• 0x01: READ</li> <li>• 0x02: WRITE</li> <li>• 0x04: WRITE_WO_RESP</li> <li>• 0x08: SIGNED_WRITE</li> </ul>
GATT_Evt_Mask	1	GATT event mask	<p>Bitmask of:</p> <ul style="list-style-type: none"> <li>• 0x00: GATT_DONT_NOTIFY_EVENTS</li> <li>• 0x01: GATT_NOTIFY_ATTRIBUTE_WRITE</li> <li>• 0x02: GATT_NOTIFY_WRITE_REQ_AND_WAIT_FOR_APPL_RESP</li> <li>• 0x04: GATT_NOTIFY_READ_REQ_AND_WAIT_FOR_APPL_RESP</li> <li>• 0x08: GATT_NOTIFY_NOTIFICATION_COMPLETION</li> </ul>
Enc_Key_Size	1	Minimum encryption key size required to read the characteristic	0x07 ... 0x10
Is_Variable	1	Specify if the characteristic value has a fixed or a variable length	<ul style="list-style-type: none"> <li>• 0x00: Fixed length</li> <li>• 0x01: Variable length</li> </ul>

Table 279. ACI\_GATT\_ADD\_CHAR\_DESC output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-
Char_Desc_Handle	2	Handle of the characteristic descriptor	-

**Events generated**

- HCI\_COMMAND\_COMPLETE\_EVENT

**2.5.6****ACI\_GATT\_UPDATE\_CHAR\_VALUE**

Updates a characteristic value in a service. If notifications (or indications) are enabled on that characteristic, a notification (or indication) is sent to the client after sending this command to the MCU. The command is queued into the STM32WB command queue. If the buffer is full (because previous commands have not been processed yet), the function returns BLE\_STATUS\_INSUFFICIENT\_RESOURCES. This happens if notifications (or indications) are enabled and the application calls ACI\_GATT\_UPDATE\_CHAR\_VALUE at a rate higher than the one allowed by the link. Throughput on BLE link depends upon connection interval and connection length parameters (decided by the central, see [ACI\\_L2CAP\\_CONNECTION\\_PARAMETER\\_UPDATE\\_REQ](#) for more info on how to suggest new connection parameters from a peripheral). If the application does not want to lose notifications because the MCU buffer becomes full, it must retry until the function returns BLE\_STATUS\_SUCCESS, or any other error code.

Note that the characteristic is updated only if the command returns BLE\_STATUS\_SUCCESS or BLE\_STATUS\_SEC\_PERMISSION\_ERROR (0x65). The security permission error means that at least one client has not been notified due to security requirements not met.

**Table 280.** ACI\_GATT\_UPDATE\_CHAR\_VALUE input parameters

Parameter	Size	Description	Possible values
Service_Handle	2	Handle of service to whom the characteristic belongs	-
Char_Handle	2	Handle of the characteristic declaration	-
Val_Offset	1	The offset from which the attribute value must be updated. If this is set to 0 and the attribute value is of variable length, then the length of the attribute is set to the Char_Value_Length. If the Val_Offset is set to a value greater than 0, then the length of the attribute is set to the maximum length as specified for the attribute while adding the characteristic.	-
Char_Value_Length	1	Length of the characteristic value in octets	-
Char_Value	Char_Value_Length	Characteristic value	-

**Table 281.** ACI\_GATT\_UPDATE\_CHAR\_VALUE output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- HCI\_COMMAND\_COMPLETE\_EVENT
- ACI\_GATT\_NOTIFICATION\_COMPLETE\_EVENT

**2.5.7 ACI\_GATT\_DEL\_CHAR**

Deletes the specified characteristic from the service.

**Table 282.** ACI\_GATT\_DEL\_CHAR input parameters

Parameter	Size	Description	Possible values
Serv_Handle	2	Handle of service to whom the characteristic belongs	-
Char_Handle	2	Handle of the characteristic to be deleted	-

**Output parameters****Table 283.** ACI\_GATT\_DEL\_CHAR output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- HCI\_COMMAND\_COMPLETE\_EVENT

**2.5.8 ACI\_GATT\_DEL\_SERVICE**

Deletes the specified service from the GATT server database.

**Table 284.** ACI\_GATT\_DEL\_SERVICE input parameters

Parameter	Size	Description	Possible values
Serv_Handle	2	Handle of the service to be deleted	-

**Table 285. ACI\_GATT\_DEL\_SERVICE output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.5.9 ACI\_GATT\_DEL\_INCLUDE\_SERVICE**

Deletes the include definition from the service.

**Table 286. ACI\_GATT\_DEL\_INCLUDE\_SERVICE input parameters**

Parameter	Size	Description	Possible values
Serv_Handle	2	Handle of the service to whom the included service belongs	-
Include_Handle	2	Handle of the included service to be deleted	-

**Table 287. ACI\_GATT\_DEL\_INCLUDE\_SERVICE output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.5.10 ACI\_GATT\_SET\_EVENT\_MASK**

Masks events from the GATT. If the bit in GATT\_Evt\_Mask is set to 1, the event associated with that bit is enabled. In the default configuration all the events are masked.

**Table 288. ACI\_GATT\_SET\_EVENT\_MASK input parameters**

Parameter	Size	Description	Possible values
GATT_Evt_Mask	4	GATT/ATT event mask	<ul style="list-style-type: none"> <li>• 0x00000001: ACI_GATT_ATTRIBUTE_MODIFIED_EVENT</li> <li>• 0x00000002: ACI_GATT_PROC_TIMEOUT_EVENT</li> <li>• 0x00000004: ACI_ATT_EXCHANGE_MTU_RESP_EVENT</li> <li>• 0x00000008: ACI_ATT_FIND_INFO_RESP_EVENT</li> <li>• 0x00000010: ACI_ATT_FIND_BY_TYPE_VALUE_RESP_EVENT</li> <li>• 0x00000020: ACI_ATT_READ_BY_TYPE_RESP_EVENT</li> <li>• 0x00000040: ACI_ATT_READ_RESP_EVENT</li> <li>• 0x00000080: ACI_ATT_READ_BLOB_RESP_EVENT</li> <li>• 0x00000100: ACI_ATT_READ_MULTIPLE_RESP_EVENT</li> <li>• 0x00000200: ACI_ATT_READ_BY_GROUP_TYPE_RESP_EVENT</li> <li>• 0x00000800: ACI_ATT_PREPARE_WRITE_RESP_EVENT</li> <li>• 0x00001000: ACI_ATT_EXEC_WRITE_RESP_EVENT</li> <li>• 0x00002000: ACI_GATT_INDICATION_EVENT</li> <li>• 0x00004000: ACI_GATT_NOTIFICATION_EVENT</li> <li>• 0x00008000: ACI_GATT_ERROR_RESP_EVENT</li> <li>• 0x00010000: ACI_GATT_PROC_COMPLETE_EVENT</li> <li>• 0x00020000: ACI_GATT_DISC_READ_CHAR_BY_UUID_RESP_EVENT</li> <li>• 0x00040000: ACI_GATT_TX_POOL_AVAILABLE_EVENT</li> <li>• 0x00100000: ACI_GATT_READ_EXT_EVENT</li> <li>• 0x00200000: ACI_GATT_INDICATION_EXT_EVENT</li> <li>• 0x00400000: ACI_GATT_NOTIFICATION_EXT_EVENT</li> </ul>

**Table 289. ACI\_GATT\_SET\_EVENT\_MASK output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.5.11 ACI\_GATT\_EXCHANGE\_CONFIG**

Performs an ATT MTU exchange procedure. When the ATT MTU exchange procedure is completed, an [ACI\\_ATT\\_EXCHANGE\\_MTU\\_RESP\\_EVENT](#) is generated. An [ACI\\_GATT\\_PROC\\_COMPLETE\\_EVENT](#) is also generated to indicate the end of the procedure.

**Table 290. ACI\_GATT\_EXCHANGE\_CONFIG input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle for which the command is given	0x0000 ... 0x0EFF

**Table 291. ACI\_GATT\_EXCHANGE\_CONFIG output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_STATUS\\_EVENT](#)
- [ACI\\_ATT\\_EXCHANGE\\_MTU\\_RESP\\_EVENT](#)

**2.5.12 ACI\_ATT\_FIND\_INFO\_REQ**

Sends a find information request. This command is used to obtain the mapping of attribute handles with their associated types. The responses of the procedure are given through the [ACI\\_ATT\\_FIND\\_INFO\\_RESP\\_EVENT](#). The end of the procedure is indicated by an [ACI\\_GATT\\_PROC\\_COMPLETE\\_EVENT](#).

**Table 292. ACI\_ATT\_FIND\_INFO\_REQ input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle for which the command is given	0x0000 ... 0x0EFF
Start_Handle	2	First requested handle number	-
End_Handle	2	Last requested handle number	-

**Table 293. ACI\_ATT\_FIND\_INFO\_REQ output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_STATUS\\_EVENT](#)
- [ACI\\_ATT\\_FIND\\_INFO\\_RESP\\_EVENT](#)
- [ACI\\_GATT\\_PROC\\_COMPLETE\\_EVENT](#)

## 2.5.13

**ACI\_ATT\_FIND\_BY\_TYPE\_VALUE\_REQ**

Sends a find by type value request, used to obtain the handles of attributes that have a given 16-bit UUID attribute type and a given attribute value. The responses of the procedure are given through the ACI\_ATT\_FIND\_BY\_TYPE\_VALUE\_RESP\_EVENT. The end of the procedure is indicated by an ACI\_GATT\_PROC\_COMPLETE\_EVENT.

**Table 294. ACI\_ATT\_FIND\_BY\_TYPE\_VALUE\_REQ input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle for which the command is given	0x0000 ... 0x0EFF
Start_Handle	2	First requested handle number	-
End_Handle	2	Last requested handle number	-
UUID	2	2 octet UUID to find (little-endian)	-
Attribute_Val_Length	1	Length of attribute value (maximum value is ATT_MTU - 7).	-
Attribute_Val	Attribute_Val_Length	Attribute value to find	-

**Table 295. ACI\_ATT\_FIND\_BY\_TYPE\_VALUE\_REQ output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- HCI\_COMMAND\_STATUS\_EVENT
- ACI\_ATT\_FIND\_INFO\_RESP\_EVENT
- ACI\_GATT\_ERROR\_RESP\_EVENT
- ACI\_GATT\_PROC\_COMPLETE\_EVENT

## 2.5.14

**ACI\_ATT\_READ\_BY\_TYPE\_REQ**

Sends a read by type request. The read by type request is used to obtain the values of attributes where the attribute type is known, but the handle is not. The responses are given through the ACI\_ATT\_READ\_BY\_TYPE\_RESP\_EVENT.

**Table 296. ACI\_ATT\_READ\_BY\_TYPE\_REQ input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle for which the command is given	0x0000 ... 0x0EFF
Start_Handle	2	First requested handle number	-
End_Handle	2	Last requested handle number	-
UUID_Type	1	UUID type: 0x01 = 16 bits UUID while 0x02 = 128 bits UUID	-
UUID	2 or 16	16- or 128-bit UUID	-

**Table 297. ACI\_ATT\_READ\_BY\_TYPE\_REQ output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- HCI\_COMMAND\_STATUS\_EVENT
- ACI\_ATT\_READ\_BY\_TYPE\_RESP\_EVENT

- ACI\_GATT\_ERROR\_RESP\_EVENT
- ACI\_GATT\_PROC\_COMPLETE\_EVENT

### 2.5.15 ACI\_ATT\_READ\_BY\_GROUP\_TYPE\_REQ

Sends a read by group type request, used to obtain the values of grouping attributes where the attribute type is known but the handle is not. Grouping attributes are defined at GATT layer. The grouping attribute types are: "Primary Service", "Secondary Service" and "Characteristic". The responses of the procedure are given through the ACI\_ATT\_READ\_BY\_GROUP\_TYPE\_RESP\_EVENT. The end of the procedure is indicated by an ACI\_GATT\_PROC\_COMPLETE\_EVENT.

**Table 298. ACI\_ATT\_READ\_BY\_GROUP\_TYPE\_REQ input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle for which the command is given	0x0000 ... 0x0EFF
Start_Handle	2	First requested handle number	-
End_Handle	2	Last requested handle number	-
UUID_Type	1	UUID type: 0x01 = 16 bits UUID while 0x02 = 128 bits UUID	-
UUID	2 or 16	16- or 128-bit UUID	-

**Table 299. ACI\_ATT\_READ\_BY\_GROUP\_TYPE\_REQ output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

#### Events generated

- HCI\_COMMAND\_STATUS\_EVENT
- ACI\_ATT\_READ\_BY\_GROUP\_TYPE\_RESP\_EVENT
- ACI\_GATT\_ERROR\_RESP\_EVENT
- ACI\_GATT\_PROC\_COMPLETE\_EVENT

### 2.5.16 ACI\_ATT\_PREPARE\_WRITE\_REQ

Sends a prepare write request, used to request the server to prepare to write the value of an attribute. The responses of the procedure are given through the ACI\_ATT\_PREPARE\_WRITE\_RESP\_EVENT. The end of the procedure is indicated by an ACI\_GATT\_PROC\_COMPLETE\_EVENT.

**Table 300. ACI\_ATT\_PREPARE\_WRITE\_REQ input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle for which the command is given	0x0000 ... 0x0EFF
Attr_Handle	2	Handle of the attribute to be written	-
Val_Offset	2	The offset of the first octet to be written	-
Attribute_Val_Length	1	Length of attribute value (maximum value is ATT_MTU - 5)	-
Attribute_Val	Attribute_Val_Length	The value of the attribute to be written	-

**Table 301. ACI\_ATT\_PREPARE\_WRITE\_REQ output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- HCI\_COMMAND\_STATUS\_EVENT
- ACI\_ATT\_PREPARE\_WRITE\_RESP\_EVENT
- ACI\_GATT\_ERROR\_RESP\_EVENT
- ACI\_GATT\_PROC\_COMPLETE\_EVENT

**2.5.17 ACI\_ATT\_EXECUTE\_WRITE\_REQ**

Sends an execute write request. The execute write request is used to request the server to write or cancel the write of all the prepared values currently held in the prepare queue from this client. The result of the procedure is given through the ACI\_ATT\_EXEC\_WRITE\_RESP\_EVENT. The end of the procedure is indicated by an ACI\_GATT\_PROC\_COMPLETE\_EVENT.

**Table 302. ACI\_ATT\_EXECUTE\_WRITE\_REQ input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle for which the command is given	0x0000 ... 0x0EFF
Execute	1	Execute or cancel writes.	<ul style="list-style-type: none"><li>• 0x00: Cancel all prepared writes</li><li>• 0x01: Immediately write all pending prepared values</li></ul>

**Table 303. ACI\_ATT\_EXECUTE\_WRITE\_REQ output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- HCI\_COMMAND\_STATUS\_EVENT
- ACI\_ATT\_EXEC\_WRITE\_RESP\_EVENT
- ACI\_GATT\_PROC\_COMPLETE\_EVENT

**2.5.18 ACI\_GATT\_DISC\_ALL\_PRIMARY\_SERVICES**

Starts the GATT client procedure to discover all primary services on the server. The responses of the procedure are given through the ACI\_ATT\_READ\_BY\_GROUP\_TYPE\_RESP\_EVENT.

**Table 304. ACI\_GATT\_DISC\_ALL\_PRIMARY\_SERVICES input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"><li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li><li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li></ul>

**Table 305. ACI\_GATT\_DISC\_ALL\_PRIMARY\_SERVICES output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- HCI\_COMMAND\_STATUS\_EVENT
- ACI\_ATT\_READ\_BY\_GROUP\_TYPE\_RESP\_EVENT
- ACI\_GATT\_ERROR\_RESP\_EVENT
- ACI\_GATT\_PROC\_COMPLETE\_EVENT

## 2.5.19

**ACI\_GATT\_DISC\_PRIMARY\_SERVICE\_BY\_UUID**

Starts the procedure to discover the primary services of the specified UUID on the server. The responses are given through the ACI\_ATT\_FIND\_BY\_TYPE\_VALUE\_RESP\_EVENT. The end of the procedure is indicated by an ACI\_GATT\_PROC\_COMPLETE\_EVENT.

**Table 306. ACI\_GATT\_DISC\_PRIMARY\_SERVICE\_BY\_UUID input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"><li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li><li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li></ul>
UUID_Type	1	UUID type: 0x01 = 16 bits UUID while 0x02 = 128 bits UUID	-
UUID	2 or 16	16- or 128-bit UUID	-

**Output parameters****Table 307. ACI\_GATT\_DISC\_PRIMARY\_SERVICE\_BY\_UUID output parameters**

Parameter	Size	Description	Possible values
Status	1	Error code	-

**Events generated**

- HCI\_COMMAND\_COMPLETE\_EVENT
- ACI\_ATT\_FIND\_BY\_TYPE\_VALUE\_RESP\_EVENT
- ACI\_GATT\_ERROR\_RESP\_EVENT
- ACI\_GATT\_PROC\_COMPLETE\_EVENT

## 2.5.20

**ACI\_GATT\_FIND\_INCLUDED\_SERVICES**

Starts the procedure to find all included services. The responses are given through the ACI\_ATT\_READ\_BY\_TYPE\_RESP\_EVENT. The end of the procedure is indicated by an ACI\_GATT\_PROC\_COMPLETE\_EVENT.

**Table 308. ACI\_GATT\_FIND\_INCLUDED\_SERVICES input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"><li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li><li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li></ul>
Start_Handle	2	Start attribute handle of the service	-
End_Handle	2	End attribute handle of the service	-

**Output parameters****Table 309. ACI\_GATT\_FIND\_INCLUDED\_SERVICES output parameters**

Parameter	Size	Description	Possible values
Status	1	Error code	-

**Events generated**

- HCI\_COMMAND\_STATUS\_EVENT
- ACI\_ATT\_READ\_BY\_GROUP\_TYPE\_RESP\_EVENT
- ACI\_GATT\_ERROR\_RESP\_EVENT
- ACI\_GATT\_PROC\_COMPLETE\_EVENT

**2.5.21 ACI\_GATT\_DISC\_ALL\_CHAR\_OF\_SERVICE**

Starts the procedure to discover all the characteristics of a given service. When the procedure is completed, an ACI\_GATT\_PROC\_COMPLETE\_EVENT is generated. Before the completion, the response packets are given through an ACI\_ATT\_READ\_BY\_TYPE\_RESP\_EVENT.

**Table 310. ACI\_GATT\_DISC\_ALL\_CHAR\_OF\_SERVICE input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle for which the command is given	<ul style="list-style-type: none"><li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li><li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li></ul>
Start_Handle	2	Start attribute handle of the service	-
End_Handle	2	End attribute handle of the service	-

**Table 311. ACI\_GATT\_DISC\_ALL\_CHAR\_OF\_SERVICE output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- HCI\_COMMAND\_STATUS\_EVENT
- ACI\_ATT\_READ\_BY\_GROUP\_TYPE\_RESP\_EVENT
- ACI\_GATT\_ERROR\_RESP\_EVENT
- ACI\_GATT\_PROC\_COMPLETE\_EVENT

**2.5.22 ACI\_GATT\_DISC\_CHAR\_BY\_UUID**

Starts the procedure to discover all the characteristics specified by a UUID. When completed, an ACI\_GATT\_PROC\_COMPLETE\_EVENT is generated. Before completion, the response packets are given through ACI\_GATT\_DISC\_READ\_CHAR\_BY\_UUID\_RESP\_EVENT.

**Table 312. ACI\_GATT\_DISC\_CHAR\_BY\_UUID input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"><li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li><li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li></ul>
Start_Handle	2	Start attribute handle of the service	-
End_Handle	2	End attribute handle of the service	-
UUID_Type	1	UUID type: 0x01 = 16 bits UUID while 0x02 = 128 bits UUID	-
UUID	2 or 16	16- or 128-bit UUID	-

### Output parameters

Table 313. ACI\_GATT\_DISC\_CHAR\_BY\_UUID output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-

#### Events generated

- HCI\_COMMAND\_STATUS\_EVENT
- ACI\_GATT\_DISC\_READ\_CHAR\_BY\_UUID\_RESP\_EVENT
- ACI\_GATT\_ERROR\_RESP\_EVENT
- ACI\_GATT\_PROC\_COMPLETE\_EVENT

### 2.5.23 ACI\_GATT\_DISC\_ALL\_CHAR\_DESC

Starts the procedure to discover all characteristic descriptors on the server. When the procedure is completed, an ACI\_GATT\_PROC\_COMPLETE\_EVENT is generated. Before completion, the response packets are given through an ACI\_ATT\_FIND\_INFO\_RESP\_EVENT.

Table 314. ACI\_GATT\_DISC\_ALL\_CHAR\_DESC input parameters

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"><li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li><li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li></ul>
Char_Handle	2	Handle of the characteristic value	-
End_Handle	2	End handle of the characteristic	-

Table 315. ACI\_GATT\_DISC\_ALL\_CHAR\_DESC output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-

#### Events generated

- HCI\_COMMAND\_STATUS\_EVENT
- ACI\_ATT\_FIND\_INFO\_RESP\_EVENT
- ACI\_GATT\_ERROR\_RESP\_EVENT
- ACI\_GATT\_PROC\_COMPLETE\_EVENT

### 2.5.24 ACI\_GATT\_READ\_CHAR\_VALUE

Starts the procedure to read the attribute value. When completed, an ACI\_GATT\_PROC\_COMPLETE\_EVENT is generated. Before completion, the response packet is given through an ACI\_ATT\_READ\_RESP\_EVENT.

**Table 316.** ACI\_GATT\_READ\_CHAR\_VALUE input parameters

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"> <li>0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li> <li>0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li> </ul>
Attr_Handle	2	Handle of the characteristic value to be read	-

**Table 317.** ACI\_GATT\_READ\_CHAR\_VALUE output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_STATUS\\_EVENT](#)
- [ACI\\_ATT\\_READ\\_RESP\\_EVENT](#)
- [ACI\\_GATT\\_PROC\\_COMPLETE\\_EVENT](#)

**2.5.25 ACI\_GATT\_READ\_USING\_CHAR\_UUID**

This command sends a Read By Type Request packet to the server in order to read the value attribute of the characteristics specified by the UUID. When the procedure is completed, an [ACI\\_GATT\\_PROC\\_COMPLETE\\_EVENT](#) is generated. Before completion, the response packets are given through an [ACI\\_GATT\\_DISC\\_READ\\_CHAR\\_BY\\_UUID\\_RESP\\_EVENT](#).

**Note:**

*The number of bytes of a value reported by [ACI\\_GATT\\_DISC\\_READ\\_CHAR\\_BY\\_UUID\\_RESP\\_EVENT](#) cannot exceed [BLE\\_EVT\\_MAX\\_PARAM\\_LEN](#) - 7 (248 bytes for default value of [BLE\\_EVT\\_MAX\\_PARAM\\_LEN](#)).*

**Table 318.** ACI\_GATT\_READ\_USING\_CHAR\_UUID input parameters

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"> <li>0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li> <li>0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li> </ul>
Start_Handle	2	Starting handle of the range to be searched	-
End_Handle	2	End handle of the range to be searched	-
UUID_Type	1	UUID type: 0x01 = 16 bits UUID while 0x02 = 128 bits UUID	-
UUID	2 or 16	16- or 128-bit UUID	-

**Output parameters****Table 319.** ACI\_GATT\_READ\_USING\_CHAR\_UUID output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_STATUS\\_EVENT](#)

- ACI\_GATT\_DISC\_READ\_CHAR\_BY\_UUID\_RESP\_EVENT
- ACI\_GATT\_ERROR\_RESP\_EVENT
- ACI\_GATT\_PROC\_COMPLETE\_EVENT

### 2.5.26 ACI\_GATT\_READ\_LONG\_CHAR\_VALUE

Start the procedure to read a long characteristic value. When the procedure is completed, an ACI\_GATT\_PROC\_COMPLETE\_EVENT is generated. Before completion, the response packets are given through an ACI\_ATT\_READ\_BLOB\_RESP\_EVENT.

**Table 320. ACI\_GATT\_READ\_LONG\_CHAR\_VALUE input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"><li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li><li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li></ul>
Attr_Handle	2	Handle of the characteristic value to be read	-
Val_Offset	2	Offset from which the value must be read	-

**Table 321. ACI\_GATT\_READ\_LONG\_CHAR\_VALUE output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

#### Events generated

- HCI\_COMMAND\_STATUS\_EVENT
- ACI\_ATT\_READ\_BLOB\_RESP\_EVENT
- ACI\_GATT\_ERROR\_RESP\_EVENT
- ACI\_GATT\_PROC\_COMPLETE\_EVENT

### 2.5.27 ACI\_GATT\_READ\_MULTIPLE\_CHAR\_VALUE

Starts a procedure to read multiple characteristic values from a server. This sub-procedure is used to read multiple characteristic values from a server when the client knows the characteristic value handles. When the procedure is completed, an ACI\_GATT\_PROC\_COMPLETE\_EVENT is generated. Before completion, the response packets are given through an ACI\_ATT\_READ\_MULTIPLE\_RESP\_EVENT.

**Table 322. ACI\_GATT\_READ\_MULTIPLE\_CHAR\_VALUE input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"><li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li><li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li></ul>
Number_of_Handles	1	The number of handles for which the value must be read	-
Handle[i]	2	The handles for which the attribute value must be read	-

**Table 323. ACI\_GATT\_READ\_MULTIPLE\_CHAR\_VALUE output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- HCI\_COMMAND\_STATUS\_EVENT
- ACI\_GATT\_ERROR\_RESP\_EVENT
- ACI\_GATT\_PROC\_COMPLETE\_EVENT

**2.5.28****ACI\_GATT\_WRITE\_CHAR\_VALUE**

Starts the procedure to write a characteristic value. When the procedure is completed, an ACI\_GATT\_PROC\_COMPLETE\_EVENT is generated.

**Table 324. ACI\_GATT\_WRITE\_CHAR\_VALUE input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle for which the command is given	<ul style="list-style-type: none"> <li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li> <li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li> </ul>
Attr_Handle	2	Handle of the characteristic value to be written	-
Attribute_Val_Length	1	Length of the value to be written	-
Attribute_Val	Attribute_Val_Length	Value to be written	-

**Table 325. ACI\_GATT\_WRITE\_CHAR\_VALUE output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- HCI\_COMMAND\_STATUS\_EVENT
- ACI\_GATT\_PROC\_COMPLETE\_EVENT
- ACI\_GATT\_ERROR\_RESP\_EVENT

**2.5.29****ACI\_GATT\_WRITE\_LONG\_CHAR\_VALUE**

Starts the procedure to write a long characteristic value. When the procedure is completed, an ACI\_GATT\_PROC\_COMPLETE\_EVENT is generated. During the procedure, ACI\_ATT\_PREPARE\_WRITE\_RESP\_EVENT and ACI\_ATT\_EXEC\_WRITE\_RESP\_EVENT are raised.

**Table 326.** ACI\_GATT\_WRITE\_LONG\_CHAR\_VALUE input parameters

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"> <li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li> <li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li> </ul>
Attr_Handle	2	Handle of the characteristic value to be written	-
Val_Offset	2	Offset at which the attribute must be written	-
Attribute_Val_Length	1	Length of the value to be written	-
Attribute_Val	Attribute_Val_Length	Value to be written	-

**Table 327.** ACI\_GATT\_WRITE\_LONG\_CHAR\_VALUE output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_STATUS\\_EVENT](#)
- [ACI\\_GATT\\_PROC\\_COMPLETE\\_EVENT](#)
- [ACI\\_ATT\\_PREPARE\\_WRITE\\_RESP\\_EVENT](#)
- [ACI\\_ATT\\_EXEC\\_WRITE\\_RESP\\_EVENT](#)

**2.5.30 ACI\_GATT\_WRITE\_CHAR\_RELIABLE**

Starts the procedure to write reliably a characteristic. When the procedure is completed, an [ACI\\_GATT\\_PROC\\_COMPLETE\\_EVENT](#) is generated. During the procedure, [ACI\\_ATT\\_PREPARE\\_WRITE\\_RESP\\_EVENT](#) and [ACI\\_ATT\\_EXEC\\_WRITE\\_RESP\\_EVENT](#) are raised.

**Table 328.** ACI\_GATT\_WRITE\_CHAR\_RELIABLE input parameters

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"> <li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li> <li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li> </ul>
Attr_Handle	2	Handle of the attribute to be written	-
Val_Offset	2	Offset at which the attribute has to be written	-
Attribute_Val_Length	1	Length of the value to be written	-
Attribute_Val	Attribute_Val_Length	Value to be written	-

**Table 329.** ACI\_GATT\_WRITE\_CHAR\_RELIABLE output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- HCI\_COMMAND\_STATUS\_EVENT
- ACI\_GATT\_PROC\_COMPLETE\_EVENT
- ACI\_ATT\_PREPARE\_WRITE\_RESP\_EVENT
- ACI\_ATT\_EXEC\_WRITE\_RESP\_EVENT

**2.5.31 ACI\_GATT\_WRITE\_LONG\_CHAR\_DESC**

Starts the procedure to write a long characteristic descriptor. When completed, an ACI\_GATT\_PROC\_COMPLETE\_EVENT is generated. During the procedure, ACI\_ATT\_PREPARE\_WRITE\_RESP\_EVENT and ACI\_ATT\_EXEC\_WRITE\_RESP\_EVENT are raised.

**Table 330. ACI\_GATT\_WRITE\_LONG\_CHAR\_DESC input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"><li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li><li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li></ul>
Attr_Handle	2	Handle of the attribute to be written	-
Val_Offset	2	Offset at which the attribute has to be written	-
Attribute_Val_Length	1	Length of the value to be written	-
Attribute_Val	Attribute_Val_Length	Value to be written	-

**Table 331. ACI\_GATT\_WRITE\_LONG\_CHAR\_DESC output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- HCI\_COMMAND\_STATUS\_EVENT
- ACI\_GATT\_PROC\_COMPLETE\_EVENT
- ACI\_ATT\_PREPARE\_WRITE\_RESP\_EVENT
- ACI\_ATT\_EXEC\_WRITE\_RESP\_EVENT

**2.5.32 ACI\_GATT\_READ\_LONG\_CHAR\_DESC**

Starts the procedure to read a long characteristic value. When completed, an ACI\_GATT\_PROC\_COMPLETE\_EVENT is generated. Before completion, the response packets are given through an ACI\_ATT\_READ\_BLOB\_RESP\_EVENT.

**Table 332. ACI\_GATT\_READ\_LONG\_CHAR\_DESC input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"><li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li><li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li></ul>
Attr_Handle	2	Handle of the characteristic descriptor	-
Val_Offset	2	Offset from which the value must be read	-

**Table 333. ACI\_GATT\_READ\_LONG\_CHAR\_DESC output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_STATUS\\_EVENT](#)
- [ACI\\_ATT\\_READ\\_BLOB\\_RESP\\_EVENT](#)
- [ACI\\_GATT\\_PROC\\_COMPLETE\\_EVENT](#)

**2.5.33 ACI\_GATT\_WRITE\_CHAR\_DESC**

Starts the procedure to write a characteristic descriptor. When completed, an [ACI\\_GATT\\_PROC\\_COMPLETE\\_EVENT](#) is generated.

**Table 334. ACI\_GATT\_WRITE\_CHAR\_DESC input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"><li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li><li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li></ul>
Attr_Handle	2	Handle of the attribute to be written	-
Attribute_Val_Length	1	Length of the value to be written	-
Attribute_Val	Attribute_Val_Length	Value to be written	-

**Table 335. ACI\_GATT\_WRITE\_CHAR\_DESC output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_STATUS\\_EVENT](#)
- [ACI\\_GATT\\_PROC\\_COMPLETE\\_EVENT](#)

**2.5.34 ACI\_GATT\_READ\_CHAR\_DESC**

Starts the procedure to read the specified descriptor. When completed, an [ACI\\_GATT\\_PROC\\_COMPLETE\\_EVENT](#) is generated. Before completion, the response packet is given through an [ACI\\_ATT\\_READ\\_RESP\\_EVENT](#).

**Table 336. ACI\_GATT\_READ\_CHAR\_DESC input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"><li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li><li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li></ul>
Attr_Handle	2	Handle of the descriptor to be read	-

**Table 337. ACI\_GATT\_READ\_CHAR\_DESC output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_STATUS\\_EVENT](#)
- [ACI\\_ATT\\_READ\\_RESP\\_EVENT](#)
- [ACI\\_GATT\\_PROC\\_COMPLETE\\_EVENT](#)

**2.5.35****ACI\_GATT\_WRITE\_WITHOUT\_RESP**

Starts the procedure to write a characteristic value without waiting for responses from the server. No events are generated after the command is executed.

The length of the value to be written must not exceed neither ATT\_MTU - 3, nor BLE\_EVT\_MAX\_PARAM\_LEN - 5 (250 for the BLE\_EVT\_MAX\_PARAM\_LEN default value).

**Table 338. ACI\_GATT\_WRITE\_WITHOUT\_RESP input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"><li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li><li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li></ul>
Attr_Handle	2	Handle of the characteristic value to be written	-
Attribute_Val_Length	1	Length of the value to be written	-
Attribute_Val	Attribute_Val_Length	Value to be written	-

**Table 339. ACI\_GATT\_WRITE\_WITHOUT\_RESP output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.5.36****ACI\_GATT\_SIGNED\_WRITE\_WITHOUT\_RESP**

Starts a signed write without response from the server. The procedure is used to write a characteristic value with an authentication signature, without waiting for responses from the server. It cannot be used when the link is encrypted.

The length of the value to be written must not exceed neither ATT\_MTU - 15, nor BLE\_EVT\_MAX\_PARAM\_LEN - 5 (250 for BLE\_EVT\_MAX\_PARAM\_LEN default value).

**Table 340. ACI\_GATT\_SIGNED\_WRITE\_WITHOUT\_RESP input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"> <li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li> <li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li> </ul>
Attr_Handle	2	Handle of the characteristic value to be written	-
Attribute_Val_Length	1	Length of the value to be written	-
Attribute_Val	Attribute_Val_Length	Value to be written	-

**Table 341. ACI\_GATT\_SIGNED\_WRITE\_WITHOUT\_RESP output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- `HCI_COMMAND_COMPLETE_EVENT`

**2.5.37**
**ACI\_GATT\_CONFIRM\_INDICATION**

This command must be sent when the application receives the `ACI_GATT_INDICATION_EVENT`, to confirm the indication.

**Table 342. ACI\_GATT\_CONFIRM\_INDICATION input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"> <li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li> <li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li> </ul>

**Table 343. ACI\_GATT\_CONFIRM\_INDICATION output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- `HCI_COMMAND_COMPLETE_EVENT`

**2.5.38**
**ACI\_GATT\_WRITE\_RESP**

Allows or rejects a write request from a client. This command must be sent by the application when it receives the `ACI_GATT_WRITE_PERMIT_REQ_EVENT`. If the write can be allowed, the status and error code must be set to 0. If the write cannot be allowed, the status must be set to 1, and the error code must be set to the error code to be passed to the client.

**Table 344. ACI\_GATT\_WRITE RESP input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"> <li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li> <li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li> </ul>
Attr_Handle	2	Handle of the attribute passed in the event EVT_BLUE_GATT_WRITE_PERMIT_REQ	-
Write_status	1	The value can be written or not	<ul style="list-style-type: none"> <li>• 0x00: The value can be written to the attribute specified by attr_handle</li> <li>• 0x01: The value cannot be written to the attribute specified by the attr_handle</li> </ul>
Error_Code	1	The error code passed to the client if the write must be rejected	-
Attribute_Val_Length	1	Length of the value to be written as passed in the event EVT_BLUE_GATT_WRITE_PERMIT_REQ	-
Attribute_Val	Attribute_Val_Length	Value as passed in the event EVT_BLUE_GATT_WRITE_PERMIT_REQ	-

**Table 345. ACI\_GATT\_WRITE RESP output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- HCI\_COMMAND\_COMPLETE\_EVENT

**2.5.39 ACI\_GATT\_ALLOW\_READ**

Allows the GATT server to send a response to a read request from a client. The application must send this command when it receives the ACI\_GATT\_READ\_PERMIT\_REQ\_EVENT or ACI\_GATT\_READ\_MULTI\_PERMIT\_REQ\_EVENT. The command indicates to the stack that the response can be sent to the client. If the application must update any of the attributes before they are read by the client, it must update the characteristic values using the ACI\_GATT\_UPDATE\_CHAR\_VALUE, and then issue the command. The application must perform the required operations within 30 seconds, otherwise the GATT procedure is timed out.

**Table 346. ACI\_GATT\_ALLOW\_READ input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"> <li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li> <li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li> </ul>

**Table 347. ACI\_GATT\_ALLOW\_READ output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.5.40****ACI\_GATT\_SET\_SECURITY\_PERMISSION**

This command sets the security permission for the attribute handle specified. Currently the setting of security permission is allowed only for client configuration descriptor.

**Table 348. ACI\_GATT\_SET\_SECURITY\_PERMISSION input parameters**

Parameter	Size	Description	Possible values
Serv_Handle	2	Handle of the service containing the attribute whose security permission has to be modified	-
Attr_Handle	2	Handle of the attribute whose security permission has to be modified	-
Security_Permissions	1	Security permission flags	Bitmask of: • 0x00: None • 0x01: AUTHEN_READ (needs authentication to read) • 0x02: AUTHOR_READ (needs authorization to read) • 0x04: ENCRY_READ (needs encryption to read) • 0x08: AUTHEN_WRITE (needs authentication to write) • 0x10: AUTHOR_WRITE (needs authorization to write) • 0x20: ENCRY_WRITE (needs encryption to write)

**Table 349. ACI\_GATT\_SET\_SECURITY\_PERMISSION output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.5.41****ACI\_GATT\_SET\_DESC\_VALUE**

This command sets the value of the descriptor specified by charDescHandle.

**Table 350. ACI\_GATT\_SET\_DESC\_VALUE input parameters**

Parameter	Size	Description	Possible values
Serv_Handle	2	Handle of the service containing the characteristic descriptor	-
Char_Handle	2	Handle of the characteristic containing the descriptor	-
Char_Desc_Handle	2	Handle of the descriptor whose value has to be set	-

Parameter	Size	Description	Possible values
Val_Offset	2	Offset from which the descriptor value has to be updated	-
Char_Desc_Value_Length	1	Length of the descriptor value	-
Char_Desc_Value	Char_Desc_Value_Length	Descriptor value	-

**Table 351.** ACI\_GATT\_SET\_DESC\_VALUE output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.5.42 ACI\_GATT\_READ\_HANDLE\_VALUE****Description**

Reads the value of the attribute handle specified from the local GATT database.

**Table 352.** ACI\_GATT\_READ\_HANDLE\_VALUE input parameters

Parameter	Size	Description	Possible values
Attr_Handle	2	Handle of the attribute to read	-
Offset	2	Offset from which the value needs to be read	-
Value_Length_Requested	2	Maximum number of octets to be returned as attribute value	-

**Table 353.** ACI\_GATT\_READ\_HANDLE\_VALUE output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-
Length	2	Length of the attribute value	-
Value_Length	2	Length in octets of the Value parameter	-
Value	Value_Length	Attribute value	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.5.43 ACI\_GATT\_UPDATE\_CHAR\_VALUE\_EXT**

This is a more flexible version of the ACI\_GATT\_UPDATE\_CHAR\_VALUE command. It supports the update of long attributes (up to 512 bytes), and indicates selectively the generation of indication/notification.

**Table 354.** ACI\_GATT\_UPDATE\_CHAR\_VALUE\_EXT input parameters

Parameter	Size	Description	Possible values
Conn_Handle_To_Notify	2	Specifies the client(s) to be notified.	<ul style="list-style-type: none"> <li>• 0x0000: Notify all subscribed clients on their unenhanced ATT bearer</li> <li>• 0x0001 ... 0x0EFF: Notify one client on the specified unenhanced ATT bearer (the parameter is the connection handle)</li> </ul>

Parameter	Size	Description	Possible values
			<ul style="list-style-type: none"> <li>• 0xEA00 ... 0xEA3F: Notify one client on the specified enhanced ATT bearer (the LSB of the parameter is the connection oriented channel index)</li> </ul>
Service_Handle	2	Handle of service to whom the characteristic belongs	-
Char_Handle	2	Handle of the characteristic declaration	-
Update_Type	1	Allows notification or Indication generation, if enabled in the client characteristic configuration descriptor.	Bitmask of: <ul style="list-style-type: none"> <li>• 0x00: Do not notify</li> <li>• 0x01: Notification</li> <li>• 0x02: Indication</li> </ul>
Char_Length	2	Total length of the characteristic value. For a variable size characteristic, this field specifies the new length of the characteristic value after the update; in case of fixed length, this field is ignored.	-
Value_Offset	2	The offset from which the attribute value must be updated	-
Value_Length	1	Length of the value parameter in octets	-
Value	Value_Length	Updated characteristic value	-

**Table 355. ACI\_GATT\_UPDATE\_CHAR\_VALUE\_EXT output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- HCI\_COMMAND\_COMPLETE\_EVENT
- ACI\_GATT\_NOTIFICATION\_COMPLETE\_EVENT

**2.5.44 ACI\_GATT\_DENY\_READ**

Forbids the GATT server to send a response to a read request from a client. The application can send this command when it receives the ACI\_GATT\_READ\_PERMIT\_REQ\_EVENT or ACI\_GATT\_READ\_MULTI\_PERMIT\_REQ\_EVENT. The command indicates to the stack that the client is not allowed to read the requested characteristic (for example, due to application restrictions). The error code is either 0x08 (insufficient authorization), or a value in the range 0x80-0x9F (application error). The application must issue the ACI\_GATT\_DENY\_READ or ACI\_GATT\_ALLOW\_READ command within 30 seconds from the reception of ACI\_GATT\_READ\_PERMIT\_REQ\_EVENT or ACI\_GATT\_READ\_MULTI\_PERMIT\_REQ\_EVENT, otherwise the GATT procedure issues a timeout.

**Table 356. ACI\_GATT\_DENY\_READ input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"> <li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li> <li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li> </ul>
Error_Code	1	Error code for the command	<ul style="list-style-type: none"> <li>• 0x08: Insufficient authorization</li> <li>• 0x80 ... 0x9F: application error</li> </ul>

**Table 357. ACI\_GATT\_DENY\_READ output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.5.45****ACI\_GATT\_SET\_ACCESS\_PERMISSION**

This command sets the access permission for the specified attribute handle.

**Table 358. ACI\_GATT\_SET\_ACCESS\_PERMISSION input parameters**

Parameter	Size	Description	Possible values
Serv_Handle	2	Handle of the service containing the attribute whose access permission must be modified	-
Attr_Handle	2	Handle of the attribute whose security permission must be modified	-
Access_Permissions	1	Access permission	Bitmask of: <ul style="list-style-type: none"><li>• 0x00: None</li><li>• 0x01: READ</li><li>• 0x02: WRITE</li><li>• 0x04: WRITE_WO_RESP</li><li>• 0x08: SIGNED_WRITE</li></ul>

**Table 359. ACI\_GATT\_SET\_ACCESS\_PERMISSION output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.5.46****ACI\_GATT\_STORE\_DB**

This command forces the saving of the GATT database for all active connections (by default the database is saved per active connection at the time of disconnection).

**Input parameters:** none

**Table 360. ACI\_GATT\_STORE\_DB output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.5.47****ACI\_GATT\_SET\_MULT\_NOTIFICATION**

This command sends a Multiple Handle Value Notification over the specified ATT bearer. The handles provided as parameters must be those of the characteristics declaration.

**Table 361. ACI\_GATT\_SET\_MULT\_NOTIFICATION input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"> <li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li> <li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li> </ul>
Number_Of_Handles	1	Number of handles in the next table	0x02 ... 0x7E
Handle[1]	2	Attribute handle	-

**Table 362. ACI\_GATT\_SET\_MULT\_NOTIFICATION output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

**2.5.48****ACI\_GATT\_READ\_MULTIPLE\_VAR\_CHAR\_VALUE**

Starts a procedure for reading multiple variable length characteristic values from a server. The command must specify the handles of the characteristic values to be read. When the procedure is completed, an [ACI\\_GATT\\_PROC\\_COMPLETE\\_EVENT](#) is generated. Before completion, the response packets are given through the [ACI\\_ATT\\_READ\\_MULTIPLE\\_RESP\\_EVENT](#).

**Table 363. ACI\_GATT\_READ\_MULTIPLE\_VAR\_CHAR\_VALUE input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"> <li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li> <li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li> </ul>
Number_Of_Handles	1	Number of handles in the next table	0x02 ... 0x7E
Handle[1]	2	Attribute handle	-

**Table 364. ACI\_GATT\_READ\_MULTIPLE\_VAR\_CHAR\_VALUE output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

## 2.6

## L2CAP commands

In Table 365 "Y" means that the corresponding command applies to the dedicated BLE stack variant, namely LO / BO / PO / BF / LB, respectively, for Link layer only / Beacon only / Peripheral only / Basic feature / Link layer only basic.

Table 365. L2CAP commands list

Command	OpCode	LO	PO	BO	BF	LB
ACI_L2CAP_CONNECTION_PARAMETER_UPDATE_REQ	0xFD81	-	Y	-	Y	-
ACI_L2CAP_CONNECTION_PARAMETER_UPDATE_RESP	0xFD82	-	-	-	Y	-
ACI_L2CAP_CO_C_CONNECT	0xFD88	-	-	-	-	-
ACI_L2CAP_CO_C_CONNECT_CONFIRM	0xFD89	-	-	-	-	-
ACI_L2CAP_CO_C_RECONF	0xFD8A	-	-	-	-	-
ACI_L2CAP_CO_C_RECONF_CONFIRM	0xFD8B	-	-	-	-	-
ACI_L2CAP_CO_C_DISCONNECT	0xFD8C	-	-	-	-	-
ACI_L2CAP_CO_C_FLOW_CONTROL	0xFD8D	-	-	-	-	-
ACI_L2CAP_CO_C_TX_DATA	0xFD8E	-	-	-	-	-

### 2.6.1

### ACI\_L2CAP\_CONNECTION\_PARAMETER\_UPDATE\_REQ

Sends an L2CAP connection parameter update request from the peripheral to the central. An ACI\_L2CAP\_CONNECTION\_UPDATE\_RESP\_EVENT is raised when the central responds (accepts or rejects) to the request.

Table 366. ACI\_L2CAP\_CONNECTION\_PARAMETER\_UPDATE\_REQ input parameters

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle for which the command is given	0x0000 ... 0x0EFF
Conn_Interval_Min	2	Minimum value for the connection event interval. This is less than or equal to Conn_Interval_Max. Time = N * 1.25 ms	0x0006 (7.50 ms) ... 0xC80 (4000.00 ms)
Conn_Interval_Max	2	Maximum value for the connection event interval. This is greater than or equal to Conn_Interval_Min. Time = N * 1.25 ms	0x0006 (7.50 ms) ... 0xC80 (4000.00 ms)
Conn_latency	2	Maximum peripheral latency for the connection, in number of connection events.	0x0000 ... 0x01F3
Timeout_Multiplier	2	Defines connection timeout parameter as: Timeout Multiplier * 10ms.	-

Table 367. ACI\_L2CAP\_CONNECTION\_PARAMETER\_UPDATE\_REQ output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-

#### Events generated

- HCI\_COMMAND\_STATUS\_EVENT
- ACI\_L2CAP\_CONNECTION\_UPDATE\_RESP\_EVENT

### 2.6.2

### ACI\_L2CAP\_CONNECTION\_PARAMETER\_UPDATE\_RESP

Accepts or rejects a connection update. This command must be sent in response to an ACI\_L2CAP\_CONNECTION\_UPDATE\_REQ\_EVENT from the controller. The Accept parameter must be set if the connection parameters given in the event are acceptable.

**Table 368. ACI\_L2CAP\_CONNECTION\_PARAMETER\_UPDATE\_RESP input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle for which the command is given	0x0000 ... 0x0EFF
Conn_Interval_Min	2	Minimum value for the connection event interval. This is less than or equal to Conn_Interval_Max.  Time = N * 1.25 ms	0x0006 (7.50 ms) ... 0x0C80 (4000.00 ms)
Conn_Interval_Max	2	Maximum value for the connection event interval. This is greater than or equal to Conn_Interval_Min.  Time = N * 1.25 ms	0x0006 (7.50 ms) ... 0x0C80 (4000.00 ms)
Conn_latency	2	Maximum peripheral latency for the connection, in number of connection events.	0x0000 ... 0x01F3
Timeout_Multiplier	2	Defines connection timeout parameter as: Timeout Multiplier * 10 ms	-
Minimum_CE_Length	2	Information parameter about the minimum length of connection needed for this LE connection.  Time = N * 0.625 ms	0x0000 (0.000 ms) ... 0xFFFF (40959.375 ms)
Maximum_CE_Length	2	Information parameter about the maximum length of connection needed for this LE connection.  Time = N * 0.625 ms	0x0000 (0.000 ms) ... 0xFFFF (40959.375 ms)
Identifier	1	Identifier received in ACI_L2CAP_CONNECTION_UPDATE_REQ EVENT	-
Accept	1	Specifies if connection update parameters are acceptable or not.	<ul style="list-style-type: none"> <li>• 0x00: Reject</li> <li>• 0x01: Accept</li> </ul>

**Table 369. ACI\_L2CAP\_CONNECTION\_PARAMETER\_UPDATE\_RESP output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

#### Events generated

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

#### 2.6.3 ACI\_L2CAP\_CO\_C\_CONNECT

This command sends a credit based connection request packet on the specified connection. See Bluetooth core specification Vol.3 Part A.

**Table 370. ACI\_L2CAP\_CO\_C\_CONNECT input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle to whom the command applies	0x0000 ... 0x0EFF
SPSM	2	Simplified Protocol/Service Multiplexer	0x0001 ... 0x00FF
MTU	2	Maximum transmission unit	23 ... 65535
MPS	2	Maximum payload size, in octects	23 ... 248
Initial_Credits	2	Number of K-frames that can be received on the created channel(s) by the L2CAP layer entity sending this packet	23 ... 65535
Channel_Number	1	Number of channels to be created. If this parameter is set to 0, it requests the creation of one LE credit based connection-oriented channel, otherwise it requests the creation of one or more enhanced credit based connection-oriented channels.	0 ... 5

**Table 371. ACI\_L2CAP\_CO\_C\_CONNECT output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)
- [ACI\\_L2CAP\\_CO\\_C\\_CONNECT\\_CONFIRM\\_EVENT](#)
- [ACI\\_L2CAP\\_PROC\\_TIMEOUT\\_EVENT](#)
- [ACI\\_L2CAP\\_COMMAND\\_REJECT\\_EVENT](#)

#### 2.6.4 ACI\_L2CAP\_CO\_C\_CONNECT\_CONFIRM

This command sends a credit based connection response packet. It must be used upon receipt of a connection request through an [ACI\\_L2CAP\\_CO\\_C\\_CONNECT\\_EVENT](#). See Bluetooth core specification Vol.3 Part A.

**Table 372. ACI\_L2CAP\_CO\_C\_CONNECT\_CONFIRM input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle for which the command applies	0x0000 ... 0x0EFF
MTU	2	Maximum transmission unit	23 ... 65535
MPS	2	Maximum payload size, in octects	23 ... 248
Initial_Credits	2	Number of K-frames that can be received on the created channel(s) by the L2CAP layer entity sending this packet	23 ... 65535
Results	2	This parameter indicates the outcome of the request. A value of 0x0000 indicates success, a non-zero value indicates that the request is refused.	0x0000 ... 0x000C

**Table 373. ACI\_L2CAP\_CO\_C\_CONNECT\_CONFIRM output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-
Channel_Number	1	Number of created channels (it is the length of Channel_Index_List)	0 ... 5
Channel_Index_List	Channel_Number	List of channel indexes to whom the primitive applies	-

**Events generated**

- [HCI\\_COMMAND\\_COMPLETE\\_EVENT](#)

#### 2.6.5 ACI\_L2CAP\_CO\_C\_RECONF

This command sends a credit based reconfigure request packet on the specified connection. See Bluetooth core specification Vol.3 Part A.

**Table 374. ACI\_L2CAP\_CO\_C\_RECONF input parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle to whom the command applies	0x0000 ... 0x0EFF
MTU	2	Maximum transmission unit	23 ... 65535
MPS	2	Maximum payload size in octects	23 ... 248
Channel_Number	2	Number of channels to be created (it is the length of Channel_Index_List)	1 ... 5
Channel_Index_List	Channel_Number	List of channel indexes to whom the primitive applies.	-

**Table 375.** ACI\_L2CAP\_CO\_COC\_RECONF output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- HCI\_COMMAND\_COMPLETE\_EVENT
- ACI\_L2CAP\_CO\_COC\_CONNECT\_CONFIRM\_EVENT
- ACI\_L2CAP\_PROC\_TIMEOUT\_EVENT
- ACI\_L2CAP\_COMMAND\_REJECT\_EVENT

**2.6.6****ACI\_L2CAP\_CO\_COC\_RECONF\_CONFIRM**

This command sends a credit based reconfigure response packet. It must be used upon receipt of a credit based reconfigure request through an ACI\_L2CAP\_CO\_COC\_RECONF\_EVENT. See Bluetooth core specification Vol.3 Part A.

**Table 376.** ACI\_L2CAP\_CO\_COC\_RECONF\_CONFIRM input parameters

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle to whom the command applies	0x0000 ... 0x0EFF
Results	2	This parameter indicates the outcome of the request. 0x0000 indicates success, while a non-zero value indicates that the request is refused.	0x0000 ... 0x000C

**Table 377.** ACI\_L2CAP\_CO\_COC\_RECONF\_CONFIRM output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- HCI\_COMMAND\_COMPLETE\_EVENT

**2.6.7****ACI\_L2CAP\_CO\_COC\_DISCONNECT**

This command sends a disconnection request signaling packet on the specified connection-oriented channel. See Bluetooth core specification Vol.3 Part A.

**Table 378.** ACI\_L2CAP\_CO\_COC\_DISCONNECT input parameters

Parameter	Size	Description	Possible values
Channel_Index	1	Index of the connection-oriented channel to whom the command applies	-

**Output parameters****Table 379.** ACI\_L2CAP\_CO\_COC\_DISCONNECT output parameters

Parameter	Size	Description	Possible values
Status	1	Status error code	-

**Events generated**

- HCI\_COMMAND\_COMPLETE\_EVENT
- ACI\_L2CAP\_CO\_COC\_CONNECT\_CONFIRM\_EVENT
- ACI\_L2CAP\_PROC\_TIMEOUT\_EVENT

- ACI\_L2CAP\_COMMAND\_REJECT\_EVENT

## 2.6.8 ACI\_L2CAP\_COC\_FLOW\_CONTROL

This command sends a flow control credit signaling packet on the specified connection-oriented channel. See Bluetooth core specification Vol.3 Part A.

**Table 380. ACI\_L2CAP\_COC\_FLOW\_CONTROL input parameters**

Parameter	Size	Description	Possible values
Channel_Index	1	Index of the connection-oriented channel to whom the primitive applies.	-
Credits	2	Number of credits the receiving device can increment, corresponding to the number of K-frames that can be sent to the peer device sending the Flow Control Credit packet.	1 ... 65535

**Table 381. ACI\_L2CAP\_COC\_FLOW\_CONTROL output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

### Events generated

- HCI\_COMMAND\_COMPLETE\_EVENT

## 2.6.9 ACI\_L2CAP\_COC\_TX\_DATA

This command sends a K-frame packet on the specified connection-oriented channel. See Bluetooth core specification Vol.3 Part A.

### Note:

For the first K-frame of the SDU, the information data must contain the L2CAP SDU length coded on two octets, followed by the K-frame information payload. For the next K-frames, the information data must contain only the K-frame information payload. The Length value must not exceed (BLE\_CMD\_MAX\_PARAM\_LEN - 3), i.e. 252 for BLE\_CMD\_MAX\_PARAM\_LEN default value.

**Table 382. ACI\_L2CAP\_COC\_TX\_DATA input parameters**

Parameter	Size	Description	Possible values
Channel_Index	1	Index of the connection-oriented channel to whom the primitive applies	-
Length	2	Length of data, in octects	-
Data	Length	Information data	-

**Table 383. ACI\_L2CAP\_COC\_TX\_DATA output parameters**

Parameter	Size	Description	Possible values
Status	1	Status error code	-

### Events generated

- HCI\_COMMAND\_COMPLETE\_EVENT

## 3 ACI/HCI events

### 3.1 HCI events

In Table 384 "Y" means that the corresponding command applies to the dedicated BLE stack variant, namely LO / BO / PO / BF / LB, respectively, for Link layer only / Beacon only / Peripheral only / Basic feature / Link layer only basic.

Table 384. HCI events commands list

Command	OpCode	LO	PO	BO	BF	LB
HCI_DISCONNECTION_COMPLETE_EVENT	0x05	Y	Y	-	Y	Y
HCI_ENCRYPTION_CHANGE_EVENT	0x08	Y	Y	-	Y	Y
HCI_READ_REMOTE_VERSION_INFORMATION_COMPLETE_EVENT	0x0C	Y	Y	-	Y	Y
HCI_HARDWARE_ERROR_EVENT	0x10	Y	Y	Y	Y	Y
HCI_NUMBER_OF_COMPLETED_PACKETS_EVENT	0x13	Y	-	-	-	Y
HCI_ENCRYPTION_KEY_REFRESH_COMPLETE_EVENT	0x30	Y	Y	-	Y	Y
HCI_COMMAND_COMPLETE_EVENT	0x0E	Y	Y	Y	Y	Y
HCI_COMMAND_STATUS_EVENT	0x0F	Y	Y	Y	Y	Y

#### 3.1.1 HCI\_DISCONNECTION\_COMPLETE\_EVENT

This event occurs when a connection is terminated. The status parameter indicates if the disconnection was successful or not. The Reason parameter indicates the reason for the disconnection, if the disconnection was successful. If the disconnection was not successful, it can be ignored by the host.

If the connection is terminated by the remote device, the Reason parameter is set to the reason specified by the remote device only if it has an allowed value, otherwise it is forced to Remote user terminated connection error code (0x13). Allowed remote Reason values are:

- Authentication failure error code (0x05)
- Other end terminated connection error codes (0x13 to 0x15)
- Unsupported remote feature error code (0x1A)
- Unacceptable connection parameters error code (0x3B)

Table 385. HCI\_DISCONNECTION\_COMPLETE\_EVENT parameters

Parameter	Size	Description	Possible values
Status	1	Error code	-
Connection_Handle	2	Connection handle for which the command is given	0x0000 ... 0x0EFF
Reason	1	Reason for disconnection (see Bluetooth core specification [Vol 2] Part D, Error codes)	-

#### 3.1.2 HCI\_ENCRYPTION\_CHANGE\_EVENT

This event is used to indicate that the change of the encryption mode has been completed.

The Connection\_Handle is for an ACL connection. The Encryption\_Enabled event parameter specifies the new Encryption\_Enabled parameter for the Connection\_Handle specified by the event parameter. This event occurs on both devices to notify the hosts when encryption has changed for the specified Connection\_Handle between two devices. The Encryption change event is used to indicate that the change of the encryption mode has been completed. The Connection\_Handle is a Connection\_Handle for an ACL connection. The Encryption\_Enabled event parameter specifies the new Encryption\_Enabled parameter for the Connection\_Handle specified by the Connection\_Handle event parameter. This event occurs on both devices to notify the hosts when encryption has changed for the specified Connection\_Handle between two devices.

Note: *This event is not generated if encryption is paused or resumed (for example, during a role switch).*

The meaning of the Encryption\_Enabled parameter depends on whether the host has indicated support for secure connections in the Secure\_Connections\_Host\_Support parameter. When Secure\_Connections\_Host\_Support is 'disabled' or the Connection\_Handle refers to an LE link, the controller uses only Encryption\_Enabled values 0x00 (OFF) and 0x01 (ON).

**Table 386. HCI\_ENCRYPTION\_CHANGE\_EVENT parameters**

Parameter	Size	Description	Possible values
Status	1	Error code	-
Connection_Handle	2	Connection handle for which the command is given	0x0000 ... 0x0EFF
Encryption_Enabled	1	Link level encryption	<ul style="list-style-type: none"><li>• 0x00: Link level encryption OFF</li><li>• 0x01: Link level encryption is ON with AES-CCM</li></ul>

### 3.1.3 HCI\_READ\_REMOTE\_VERSION\_INFORMATION\_COMPLETE\_EVENT

This event is used to indicate the completion of the process obtaining the version information of the remote controller specified by the Connection\_Handle event parameter.

The Connection\_Handle is for an ACL connection. The version event parameter defines the specification version of the LE controller. The Manufacturer\_Name event parameter indicates the manufacturer of the remote controller. The subversion event parameter is controlled by the manufacturer and is implementation dependent. The subversion event parameter defines the various revisions that each version of the Bluetooth hardware goes through as design processes changes and errors are fixed. This allows the software to determine what Bluetooth hardware is being used and, if necessary, to work around various bugs in the hardware. When the Connection\_Handle is associated with an LE-U logical link, the version event parameter is link layer VersNr parameter, the Manufacturer\_Name event parameter is the compld parameter, and the subversion event parameter is the SubVersNr parameter.

**Table 387. HCI\_READ\_REMOTE\_VERSION\_INFORMATION\_COMPLETE\_EVENT parameters**

Parameter	Size	Description	Possible values
Status	1	Error code	-
Connection_Handle	2	Connection handle for which the command is given	0x0000 ... 0x0EFF
Version	1	Version of the current LMP in the remote controller	-
Manufacturer_Name	2	Manufacturer name of the remote controller	-
Subversion	2	Subversion of the LMP in the remote controller	-

### 3.1.4 HCI\_HARDWARE\_ERROR\_EVENT

This event is used to indicate the implementation specific type of hardware failure for the controller. It notifies the host that a hardware failure has occurred in the controller.

**Table 388. HCI\_HARDWARE\_ERROR\_EVENT parameters**

Parameter	Size	Description	Possible values
Hardware_Code	1	Hardware error event code: <ul style="list-style-type: none"><li>• 0: not used</li><li>• 1: bluecore act2 error</li><li>• 2: bluecore time overrun error</li><li>• 3: internal FIFO full</li><li>• 4: ISR delay error</li></ul>	<ul style="list-style-type: none"><li>• 0x00: not used</li><li>• 0x01: event_act2 error</li><li>• 0x02: event_time_overrun error</li><li>• 0x03: event_fifo_full error</li><li>• 0x04: event_isr_delay error</li></ul>

## 3.1.5

**HCI\_NUMBER\_OF\_COMPLETED\_PACKETS\_EVENT**

This event is used by the controller to indicate to the host how many HCI data packets have been completed (transmitted or flushed) for each Connection\_Handle since the previous number of completed packets event was sent to the host. This means that the corresponding buffer space has been freed in the controller. Based on this information, and on the HC\_Total\_Num\_ACL\_Data\_Packets and HC\_Total\_Num\_Synchronous\_- Data\_Packets return parameter of the Read\_Buffer\_Size command, the host determines for which Connection\_Handles the following HCI data packets must be sent to the controller. The number of completed packets event must not be sent before the corresponding connection complete event. While the controller has HCI data packets in its buffer, it must keep sending the number of completed packets event to the host at least periodically, until it finally reports that all the pending ACL data packets have been transmitted or flushed.

**Table 389. HCI\_NUMBER\_OF\_COMPLETED\_PACKETS\_EVENT parameters**

Parameter	Size	Description	Possible values
Number_of_Handles	1	The number of Connection_Handles and Num_HCI_Data_Packets parameters pairs contained in this event	-
Connection_Handle[i]	2	Connection handle	-
HC_Num_Of_Completed_Packets[i]	2	The number of HCI data packets that have been completed (transmitted or flushed) for the associated Connection_Handle since the previous time the event was returned	-

## 3.1.6

**HCI\_ENCRYPTION\_KEY\_REFRESH\_COMPLETE\_EVENT**

This event is used to indicate to the Host that the encryption key was refreshed on the given Connection\_Handle. The controller sends this event when the encryption key has been refreshed, because the encryption started or resumed.

**Table 390. HCI\_ENCRYPTION\_KEY\_REFRESH\_COMPLETE\_EVENT parameters**

Parameter	Size	Description	Possible values
Status	1	Error code	-
Connection_Handle	2	Connection handle for which the command is given	0x0000 ... 0x0EFF

## 3.1.7

**HCI\_COMMAND\_COMPLETE\_EVENT**

This event is used by the controller for most commands to transmit return status of a command and the other event parameters that are specified for the issued HCI command. The Num\_HCI\_Command\_Packets event parameter allows the controller to indicate the number of HCI command packets that the host can send to the controller. If the controller requires the host to stop sending commands, the Num\_HCI\_Command\_Packets event parameter is set to 0. To indicate to the host that the controller is ready to receive HCI command packets, the controller generates a command complete event with the Command\_Opcode 0x0000, and the Num\_HCI\_Command\_Packets event parameter is set to 1 or more. See each command for the parameters that are returned by this event.

**Table 391. HCI\_COMMAND\_COMPLETE\_EVENT parameters**

Parameter	Size	Description	Possible values
Num_HCI_Command_Packets	1	The number of HCI command packets allowed to be sent to the controller from the host.	-
Command_Opcode	2	Opcode of the command which caused this event	-
Return_Parameters	Variable	Return parameter(s) for the command specified in the Command_Opcode event parameter.	-

### 3.1.8 HCI\_COMMAND\_STATUS\_EVENT

This event is used to indicate that the command described by the Command\_Opcode parameter has been received, and that the controller is currently performing the task for this command. This event is needed to provide mechanisms for asynchronous operation, which makes it possible to prevent the host from waiting for a command to finish. If the command cannot begin to execute (a parameter error may have occurred, or the command may currently not be allowed), the status event parameter contains the corresponding error code, and no complete event follows.

The Num\_HCI\_Command\_Packets event parameter allows the controller to indicate the number of HCI command packets the host can send to the controller. If the controller requires the host to stop sending commands, this parameter is set to 0. To indicate to the host that the controller is ready to receive HCI command packets, the controller generates a command status event with status 0x00 and Command\_Opcode 0x0000, and the Num\_HCI\_Command\_Packets event parameter is set to 1 or more.

**Table 392. HCI\_COMMAND\_STATUS\_EVENT parameters**

Parameter	Size	Description	Possible values
Status	1	Error code	-
Num_HCI_Command_Packets	1	The Number of HCI command packets which are allowed to be sent to the controller from the Host	-
Command_Opcode	2	Opcode of the command which caused this event	-

### 3.2

## HCI LE META events

In Table 393 "Y" means that the corresponding command applies to the dedicated BLE stack variant, namely LO / BO / PO / BF/ LB, respectively, for Link layer only / Beacon only / Peripheral only / Basic feature / Link layer only basic.

**Table 393. HCI LE META events command list**

Command	OpCode	LO	PO	BO	BF	LB
HCI_LE_CONNECTION_COMPLETE_EVENT	0x01	Y	Y	-	Y	Y
HCI_LE_ADVERTISING_REPORT_EVENT	0x02	Y	-	Y	Y	Y
HCI_LE_CONNECTION_UPDATE_COMPLETE_EVENT	0x03	Y	Y	-	Y	Y
HCI_LE_READ_REMOTE_FEATURES_COMPLETE_EVENT	0x04	Y	Y	-	Y	Y
HCI_LE_LONG_TERM_KEY_REQUEST_EVENT	0x05	Y	Y	-	Y	Y
HCI_LE_DATA_LENGTH_CHANGE_EVENT	0x07	Y	Y	-	Y	Y
HCI_LE_READ_LOCAL_P256_PUBLIC_KEY_COMPLETE_EVENT	0x08	Y	Y	-	Y	Y
HCI_LE_GENERATE_DHKEY_COMPLETE_EVENT	0x09	Y	-	-	-	Y
HCI_LE_ENHANCED_CONNECTION_COMPLETE_EVENT	0x0A	Y	Y	-	Y	Y
HCI_LE_DIRECTED_ADVERTISING_REPORT_EVENT	0x0B	Y	-	-	Y	Y
HCI_LE_PHY_UPDATE_COMPLETE_EVENT	0x0C	Y	-	-	Y	Y
HCI_LE_EXTENDED_ADVERTISING_REPORT_EVENT	0x0D	Y	-	-	-	-
HCI_LE_SCAN_TIMEOUT_EVENT	0x11	Y	-	-	-	-
HCI_LE_ADVERTISING_SET_TERMINATED_EVENT	0x12	Y	-	-	-	-
HCI_LE_SCAN_REQUEST_RECEIVED_EVENT	0x13	Y	-	-	-	-
HCI_LE_CHANNEL_SELECTION_ALGORITHM_EVENT	0x14	Y	Y	-	-	-

### 3.2.1

#### HCI LE CONNECTION COMPLETE EVENT

This event indicates to both the hosts forming the connection that a new connection has been created. Upon the creation, a Connection\_Handle is assigned by the controller, and passed to the host in this event. If the connection establishment fails, this event is provided to the host that issued the HCI\_LE\_CREATE\_CONNECTION command to indicates if the connection establishment failed or was successful. The Central\_Clock\_Accuracy parameter is only valid for a peripheral. On a central, this parameter is set to 0x00.

**Table 394. HCI LE CONNECTION COMPLETE EVENT parameters**

Parameter	Size	Description	Possible values
Status	1	Error code	-
Connection_Handle	2	Connection handle to be used to identify the connection with the peer device	0x0000 ... 0x0EFF
Role	1	Role of the local device in the connection	<ul style="list-style-type: none"><li>• 0x00: Central</li><li>• 0x01: Peripheral</li></ul>
Peer_Address_Type	1	The address type of the peer device	<ul style="list-style-type: none"><li>• 0x00: Public device address</li><li>• 0x01: Random device address</li></ul>
Peer_Address	6	Public device address or random device address of the peer device	-
Conn_Interval	2	Connection interval used on this connection. Time = N * 1.25 ms	0x0006 (7.50 ms) ... 0x0C80 (4000.00 ms)

Parameter	Size	Description	Possible values
Conn_Latency	2	Maximum peripheral latency for the connection, in number of connection events.	0x0000 ... 0x01F3
Supervision_Timeout	2	Supervision timeout for the LE link. It is a multiple of 10 ms, and larger than $(1 + \text{connPeripheralLatency}) * \text{connInterval} * 2$ . Time = N * 10 ms	0x000A (100 ms) ... 0x0C80 (32000)
Central_Clock_Accuracy	1	Central clock accuracy, only valid for a peripheral	<ul style="list-style-type: none"> <li>• 0x00: 500 ppm</li> <li>• 0x01: 250 ppm</li> <li>• 0x02: 150 ppm</li> <li>• 0x03: 100 ppm</li> <li>• 0x04: 75 ppm</li> <li>• 0x05: 50 ppm</li> <li>• 0x06: 30 ppm</li> <li>• 0x07: 20 ppm</li> </ul>

### 3.2.2

#### **HCI\_LE\_ADVERTISING\_REPORT\_EVENT**

Indicates that one or more Bluetooth devices have responded to an active scan, or received some information during a passive scan. The controller may queue these advertising reports, and send information from multiple devices in one LE advertising report event. In the current BLE vstack version, only one report is sent per event (num\_Reports = 1).

**Table 395. HCI\_LE\_ADVERTISING\_REPORT\_EVENT parameters**

Parameter	Size	Description	Possible values
Num_Reports	1	Number of responses in this event	0x01
Event_Type[i]	1	Type of advertising report event: 1. ADV_IND: Connectable undirected advertising 2. ADV_DIRECT_IND: Connectable directed advertising 3. ADV_SCAN_IND: Scannable undirected advertising 4. ADV_NONCONN_IND: Non connectable undirected advertising 5. SCAN_RSP: Scan response	<ul style="list-style-type: none"> <li>• 0x00: ADV_IND</li> <li>• 0x01: ADV_DIRECT_IND</li> <li>• 0x02: ADV_SCAN_IND</li> <li>• 0x03: ADV_NONCONN_IND</li> <li>• 0x04: SCAN_RSP</li> </ul>
Address_Type[i]	1	Address type	<ul style="list-style-type: none"> <li>• 0x00: Public device address</li> <li>• 0x01: Random device address</li> <li>• 0x02: Public identity address</li> <li>• 0x03: Random (static) identity address</li> </ul>
Address[i]	6	Public or random device address of the device to be connected	-
Length_Data[i]	1	Length of the Data[i] field for each device who responded	0 ... 31
Data[i]	Length_Data[i]	Length_Data[i] octets of advertising or scan response data formatted	-
RSSI[i]	1	N Size: 1 octet (signed integer), in dBm	<ul style="list-style-type: none"> <li>• 127: RSSI not available</li> <li>• -127 ... 20</li> </ul>

### 3.2.3

#### **HCI\_LE\_CONNECTION\_UPDATE\_COMPLETE\_EVENT**

This event is used to indicate that the controller process to update the connection has completed. On a peripheral, if no connection parameters are updated, this event is not issued. On a central, this event is issued if the Connection\_Update command was sent.

**Table 396. HCI\_LE\_CONNECTION\_UPDATE\_COMPLETE\_EVENT parameters**

Parameter	Size	Description	Possible values
Status	1	Error code	-
Connection_Handle	2	Connection handle to be used to identify the connection with the peer device	0x0000 ... 0x0EFF
Conn_Interval	2	Connection interval used on this connection. Time = N * 1.25 ms	0x0006 (7.50 ms) ... 0xC80 (4000.00 ms)
Conn_Latency	2	Maximum peripheral latency for the connection, in number of connection events	0x0000 ... 0x01F3
Supervision_Timeout	2	Supervision timeout for the LE link. It is a multiple of 10 ms, and larger than (1 + connPeripheralLatency) * connInterval * 2. Time = N * 10 ms	0x000A (100 ms) ... 0xC80 (32000 ms)

### 3.2.4 HCI\_LE\_READ\_REMOTE\_FEATURES\_COMPLETE\_EVENT

#### Description

This event is used to indicate the completion of the process of the controller obtaining the used features of the remote Bluetooth device specified by the Connection\_Handle event parameter.

**Table 397. HCI\_LE\_READ\_REMOTE\_FEATURES\_COMPLETE\_EVENT parameters**

Parameter	Size	Description	Possible values
Status	1	Error code	-
Connection_Handle	2	Connection handle to be used to identify the connection with the peer device	0x0000 ... 0x0EFF
LE_Features	8	Bit mask list of used LE features	-

### 3.2.5 HCI\_LE\_LONG\_TERM\_KEY\_REQUEST\_EVENT

Indicates that the central device is attempting to encrypt or re-encrypt the link, and is requesting the long term key from the host.

**Table 398. HCI\_LE\_LONG\_TERM\_KEY\_REQUEST\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle to be used to identify the connection with the peer device	0x0000 ... 0x0EFF
Random_Number	8	64-bit random number	-
Encrypted_Diversifier	2	16-bit encrypted diversifier	-

### 3.2.6 HCI\_LE\_DATA\_LENGTH\_CHANGE\_EVENT

Notifies the host of a change in either the maximum payload length, or the maximum transmission time of packets, in both directions.

The values reported are the maximum actually used on the connection following the change, except that on the LE coded PHY a packet taking up to 2704  $\mu$ s to transmit may be sent even though the corresponding parameter has a lower value.

**Table 399. HCI\_LE\_DATA\_LENGTH\_CHANGE\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle to be used to identify the connection with the peer device	0x0000 ... 0x0EFF
MaxTxOctets	2	Maximum number of payload octets in a link layer packet that the local controller sends on this connection (connEffectiveMaxTxOctets)	0x001B ... 0x00FB
MaxTxTime	2	Maximum time that the local controller takes to send a link layer packet on this connection (connEffectiveMaxTxTime)	0x0148 ... 0x4290
MaxRxOctets	2	Maximum number of payload octets in a link layer packet that the local controller expects to receive on this connection (connEffectiveMaxRxOctets)	0x001B ... 0x00FB
MaxRxTime	2	Maximum time that the local controller expects to receive a link layer packet on this connection (connEffectiveMaxRxTime)	0x0148 ... 0x4290

### 3.2.7 HCI\_LE\_READ\_LOCAL\_P256\_PUBLIC\_KEY\_COMPLETE\_EVENT

This event is generated when local P-256 key generation is complete.

**Table 400. HCI\_LE\_READ\_LOCAL\_P256\_PUBLIC\_KEY\_COMPLETE\_EVENT parameters**

Parameter	Size	Description	Possible values
Status	1	Error code	-
Local_P256_Public_Key	64	Local P-256 public key	-

### 3.2.8 HCI\_LE\_GENERATE\_DHKEY\_COMPLETE\_EVENT

This event indicates that LE Diffie-Hellman key generation has been completed by the controller.

**Table 401. HCI\_LE\_GENERATE\_DHKEY\_COMPLETE\_EVENT parameters**

Parameter	Size	Description	Possible values
Status	1	Error code	-
DHKey	32	Diffie-Hellman key	-

### 3.2.9 HCI\_LE\_ENHANCED\_CONNECTION\_COMPLETE\_EVENT

#### Description

Indicates to both the hosts that a new connection has been created. Upon its creation, a Connection\_Handle is assigned by the controller, and passed to the host. If the connection set-up fails, this event is provided to the host that issued the LE\_Create\_Connection command. If this event and the LE connection complete events are unmasked, only the LE enhanced connection complete event is sent when a new connection has been completed. This event indicates to the host that issued a LE\_Create\_Connection command and received a command status event if the connection set-up failed or was successful. The Central\_Clock\_Accuracy parameter is only valid for a peripheral. On a central, this parameter is set to 0x00.

**Table 402. HCI\_LE\_ENHANCED\_CONNECTION\_COMPLETE\_EVENT parameters**

Parameter	Size	Description	Possible values
Status	1	Error code	-
Connection_Handle	2	Connection handle to be used to identify the connection with the peer device	0x0000 ... 0x0EFF
Role	1	Role of the local device in the connection	<ul style="list-style-type: none"> <li>• 0x00: Central</li> <li>• 0x01: Peripheral</li> </ul>
Peer_Address_Type	1	0x00 Public device address 0x01 Random device address	<ul style="list-style-type: none"> <li>• 0x00: Public device address</li> </ul>

Parameter	Size	Description	Possible values
		0x02 Public identity address (corresponds to resolved private address) 0x03 Random (static) identity address (corresponds to resolved private address)	<ul style="list-style-type: none"> <li>• 0x01: Random device address</li> <li>• 0x02: Public identity address</li> <li>• 0x03: Random (static) identity address</li> </ul>
Peer_Address	6	Public or random device address, public or random (static) identity address of the device to be connected	-
Local_Resolvable_Private_Address	6	Resolvable private address being used by the local device for this connection. This is only valid when the Own_Address_Type is set to 0x02 or 0x03. For other Own_Address_Type values, the controller returns all 0s.	-
Peer_Resolvable_Private_Address	6	Resolvable private address used by the peer device for this connection. This is only valid for Peer_Address_Type 0x02 and 0x03. For other values the controller returns all 0s.	-
Conn_Interval	2	Connection interval used on this connection. Time = N * 1.25 ms	0x0006 (7.50 ms) ... 0x0C80 (4000.00 ms)
Conn_Latency	2	Maximum peripheral latency for the connection in number of connection events	0x0000 ... 0x01F3
Supervision_Timeout	2	Supervision timeout for the LE Link. It is a multiple of 10 ms, and larger than $(1 + \text{connPeripheralLatency}) * \text{connInterval} * 2$ . Time = N * 10 ms	0x000A (100 ms) ... 0x0C80 (32000 ms)
Central_Clock_Accuracy	1	Central clock accuracy, valid for a peripheral	<ul style="list-style-type: none"> <li>• 0x00: 500 ppm</li> <li>• 0x01: 250 ppm</li> <li>• 0x02: 150 ppm</li> <li>• 0x03: 100 ppm</li> <li>• 0x04: 75 ppm</li> <li>• 0x05: 50 ppm</li> <li>• 0x06: 30 ppm</li> <li>• 0x07: 20 ppm</li> </ul>

### 3.2.10

#### HCI\_LE\_DIRECTED\_ADVERTISING\_REPORT\_EVENT

This vent indicates that directed advertisements have been received, where the advertiser is using a resolvable private address for the InitA field in the ADV\_DIRECT\_IND PDU and the Scanning\_Filter\_Policy is equal to 0x02 or 0x03, see HCI\_LE\_Set\_Scan\_Parameters. Direct\_Address\_Type and Direct\_Address identify the address the directed advertisements are being directed to. Address\_Type and Address are the address of the advertiser sending the directed advertisements.

**Table 403. HCI\_LE\_DIRECTED\_ADVERTISING\_REPORT\_EVENT parameters**

Parameter	Size	Description	Possible values
Num_Reports	1	Number of responses in this event	0x01
Event_Type[i]	1	Advertising type	0x01: connectable directed advertising (ADV_DIRECT_IND)
Address_Type[i]	1	Address type	<ul style="list-style-type: none"> <li>• 0x00: public device address</li> <li>• 0x01: random device address</li> <li>• 0x02: public identity address</li> <li>• 0x03: random (static) identity address</li> </ul>
Address[i]	6	Public or random device address, public or random (static) identity address of the advertising device	-
Direct_Address_Type[i]	1	Address type	0x01: random device address
Direct_Address[i]	6	Random device address	-
RSSI[i]	1	N Size: 1 octet (signed integer), in dBm	<ul style="list-style-type: none"> <li>• 127: RSSI not available</li> <li>• -127 ... 20</li> </ul>

### 3.2.11 HCI\_LE\_PHY\_UPDATE\_COMPLETE\_EVENT

This event is issued to indicate that the controller has changed the transmitter PHY in use, or the receiver PHY in use, or both. If an LE\_Set\_PHY command was sent and the controller determines that no PHYs change as a result, it issues this event immediately.

**Table 404. HCI\_LE\_PHY\_UPDATE\_COMPLETE\_EVENT parameters**

Parameter	Size	Description	Possible values
Status	1	Error code	-
Connection_Handle	2	Connection handle to be used to identify the connection with the peer device	0x0000 ... 0x0EFF
TX_PHY	1	Transmitter PHY in use	<ul style="list-style-type: none"> <li>• 0x01: The transmitter PHY for the connection is LE 1M</li> <li>• 0x02: The transmitter PHY for the connection is LE 2M</li> <li>• 0x03: The transmitter PHY for the connection is LE coded (not supported by STM32WB)</li> </ul>
RX_PHY	1	Receiver PHY in use	<ul style="list-style-type: none"> <li>• 0x01: The receiver PHY for the connection is LE 1M</li> <li>• 0x02: The receiver PHY for the connection is LE 2M</li> <li>• 0x03: The receiver PHY for the connection is LE coded (not supported by STM32WB)</li> </ul>

### 3.2.12 HCI\_LE\_EXTENDED\_ADVERTISING\_REPORT\_EVENT

This event indicates that a Bluetooth device has responded to an active scan or has broadcast advertisements received during a passive scan. See Bluetooth spec. v.5.2 [Vol 4, Part E, 7.7.65.13].

**Table 405. HCI\_LE\_EXTENDED\_ADVERTISING\_REPORT\_EVENT parameters**

Parameter	Size	Description	Possible values
Num_Reports	1	Number of responses	0x01
Event_Type	1	Event type	Bitmask of: <ul style="list-style-type: none"> <li>• 0x0001: connectable advertising</li> <li>• 0x0002: scannable advertising</li> <li>• 0x0004: directed advertising</li> <li>• 0x0008: scan response</li> <li>• 0x0010: legacy advertising</li> </ul>

Parameter	Size	Description	Possible values
			PDUs used: <ul style="list-style-type: none"> <li>• 0x0020: incomplete, more data to come</li> <li>• 0x0040: incomplete, data truncated, no more to come</li> </ul>
Address_Type	1	Address type of the advertising device	<ul style="list-style-type: none"> <li>• 0x00: public device address</li> <li>• 0x01: random device address</li> <li>• 0x02: public identity address (corresponds to resolved private address)</li> <li>• 0x03: random (static) identity address (corresponds to resolved private address)</li> <li>• 0xFF: no address provided (anonymous advertisement)</li> </ul>
Address	6	Public, random, public or random (static) identity address of the advertising device	-
Primary_PHY	1	Primary advertising PHY	<ul style="list-style-type: none"> <li>• 0x01: advertiser PHY is LE 1M</li> </ul>
Secondary_PHY	1	Secondary advertising PHY	<ul style="list-style-type: none"> <li>• 0x00: no packets on the secondary advertising physical channel</li> <li>• 0x01: advertiser PHY is LE 1M</li> <li>• 0x02: advertiser PHY is LE 2M</li> <li>• 0x01: advertiser PHY is LE coded</li> </ul>
Advertising_SID	1	Value of the Advertising SID subfield in the ADI field of the PDU or, for scan responses, in the ADI field of the original scannable.	<ul style="list-style-type: none"> <li>• 0xFF: no ADI field provided</li> <li>• 0x00 ... 0x0F: advertising SID subfield</li> </ul>
TX_Power	1	Tx Power (signed integer), in dBm	<ul style="list-style-type: none"> <li>• 127: Tx power information not available</li> <li>• -127 ... 20: Tx power</li> </ul>
RSSI	1	RSSI (signed integer), in dBm	<ul style="list-style-type: none"> <li>• 127: RSSI not available</li> <li>• -127 ... 20: Tx power</li> </ul>
Periodic_Adv_Interval	2	Interval of the periodic advertising	<ul style="list-style-type: none"> <li>• 0x0000: no periodic advertising</li> </ul>
Direct_Address_Type	1	Target device address type	<ul style="list-style-type: none"> <li>• 0x00: public device address</li> <li>• 0x01: random device address</li> <li>• 0x02: public identity address (corresponds to resolved private address)</li> <li>• 0x03: random (static) identity address (corresponds to resolved private address)</li> <li>• 0xFE: random device address (controller unable to resolve)</li> </ul>
Direct_Address	6	Public, random, public or random (static) identity address of the advertising device	-
Data_Length	1	Length of data	-
Data	Data_Length	Octets of advertising or scan response data formatted as defined in Bluetooth spec. v.5.2 [Vol 3, Part C, 11]	-

### 3.2.13

### HCI\_LE\_SCAN\_TIMEOUT\_EVENT

This event indicates that scanning has ended because the duration has expired. See Bluetooth spec. v.5.2 [Vol 4, Part E, 7.7.65.17].

**Event parameters:** none

### 3.2.14 HCI\_LE\_ADVERTISING\_SET\_TERMINATED\_EVENT

This event indicates that the controller has terminated advertising in the sets specified by the Advertising\_Handle parameter. See Bluetooth spec. v.5.2 [Vol 4, Part E, 7.7.65.18].

**Table 406. HCI\_LE\_ADVERTISING\_SET\_TERMINATED\_EVENT parameters**

Parameter	Size	Description	Possible values
Status	1	Error code	-
Advertising_Handle	1	Used to identify an advertising set	0x00 ... 0xEF
Connection_Handle	2	Connection handle for which the event applies	0x0000 ... 0x0EFF
Num_Completed_Ext_Adv_Events	1	Number of completed extended advertising events transmitted by the controller	0x00 ... 0xFF

### 3.2.15 HCI\_LE\_SCAN\_REQUEST RECEIVED\_EVENT

#### Description

This event indicates that a SCAN\_REQ PDU or an AUX\_SCAN\_REQ PDU has been received by the advertiser. The request contains a device address from a scanner allowed by the advertising filter policy. The advertising set is identified by Advertising\_Handle. See Bluetooth spec. v.5.2 [Vol 4, Part E, 7.7.65.19].

**Table 407. HCI\_LE\_SCAN\_REQUEST RECEIVED\_EVENT parameters**

Parameter	Size	Description	Possible values
Advertising_Handle	1	Used to identify an advertising set	0x00 ... 0xEF
Scanner_Address_Type	1	Scanner address type	<ul style="list-style-type: none"><li>• 0x00: public device address</li><li>• 0x01: random device address</li><li>• 0x02: public identity address (corresponds to resolved private address)</li><li>• 0x03: random (static) identity address (corresponds to resolved private address)</li></ul> <p>PDUs used:</p> <ul style="list-style-type: none"><li>• 0x0020: incomplete, more data to come</li><li>• 0x0040: incomplete, data truncated, no more to come</li></ul>
Scanner_Address	6	Public, random, public or random (static) identity address of the advertising device	-

### 3.2.16 HCI\_LE\_CHANNEL\_SELECTION\_ALGORITHM\_EVENT

This event indicates which channel selection algorithm is used on a data physical channel connection. See Bluetooth spec. v.5.2 [Vol 4, Part E, 7.7.65.20].

**Table 408. HCI\_LE\_CHANNEL\_SELECTION\_ALGORITHM\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle to be used to identify the connection with the peer device	0x0000 ... 0x0EFF
Channel_Selection_Algorithm	1	LE channel selection algorithm	<ul style="list-style-type: none"><li>• 0x00: algorithm #1 is used</li><li>• 0x01: algorithm #2 is used</li></ul>

### 3.3 ACI GAP events

In Table 409 "Y" means that the corresponding command applies to the dedicated BLE stack variant, namely LO / BO / PO / BF / LB, respectively, for Link layer only / Beacon only / Peripheral only / Basic feature / Link layer only- basic.

Table 409. ACI GAP events commands list

Command	OpCode	LO	PO	BO	BF	LB
ACI_GAP_LIMITED_DISCOVERABLE_EVENT	0x0400	-	Y	-	Y	-
ACI_GAP_PAIRING_COMPLETE_EVENT	0x0401	-	Y	-	Y	-
ACI_GAP_PASS_KEY_REQ_EVENT	0x0402	-	Y	-	Y	-
ACI_GAP_AUTHORIZATION_REQ_EVENT	0x0403	-	Y	-	Y	-
ACI_GAP_PERIPHERAL_SECURITY_INITIATED_EVENT	0x0404	-	Y	-	Y	-
ACI_GAP_BOND_LOST_EVENT	0x0405	-	Y	-	Y	-
ACI_GAP_PROC_COMPLETE_EVENT	0x0407	-	Y	-	Y	-
ACI_GAP_ADDR_NOT_RESOLVED_EVENT	0x0408	-	Y	-	Y	-
ACI_GAP_NUMERIC_COMPARISON_VALUE_EVENT	0x0409	-	Y	-	Y	-
ACI_GAP_KEYPRESS_NOTIFICATION_EVENT	0x040A	-	Y	-	Y	-

#### 3.3.1 ACI\_GAP\_LIMITED\_DISCOVERABLE\_EVENT

This event is generated by the controller when the limited discoverable mode ends due to timeout (180 seconds).

**Event parameters:** none

#### 3.3.2 ACI\_GAP\_PAIRING\_COMPLETE\_EVENT

This event is generated when the pairing process has completed successfully, or a pairing procedure timeout has occurred, or the pairing has failed. This is to notify the application that there has been a pairing with a remote device so that it can take further actions, or to notify that a timeout has occurred so that the upper layer can decide to disconnect the link.

Table 410. ACI\_GAP\_PAIRING\_COMPLETE\_EVENT parameters

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle on which the pairing procedure is completed	-
Status	1	Pairing status	<ul style="list-style-type: none"><li>• 0x00: Success</li><li>• 0x01: Timeout</li><li>• 0x02: Failed</li></ul>
Reason	1	Pairing reason error code (valid in case of pairing failed status)	<ul style="list-style-type: none"><li>• 0x00:</li><li>• 0x01: PASSKEY_ENTRY_FAILED</li><li>• 0x02: OOB_NOT_AVAILABLE</li><li>• 0x03: AUTH_REQ_CANNOT_BE_MET</li><li>• 0x04: CONFIRM_VALUE_FAILED</li><li>• 0x05: PAIRING_NOT_SUPPORTED</li><li>• 0x06: INSUFF_ENCRYPTION_KEY_SIZE</li><li>• 0x07: CMD_NOT_SUPPORTED</li><li>• 0x08: UNSPECIFIED_REASON</li><li>• 0x09: VERY_EARLY_NEXT_ATTEMPT</li><li>• 0x0A: SM_INVALID_PARAMS</li><li>• 0x0B: SMP_SC_DHKEY_CHECK_FAILED</li><li>• 0x0C: SMP_SC_NUMCOMPARISON_FAILED</li></ul>

### 3.3.3 ACI\_GAP\_PASS\_KEY\_REQ\_EVENT

This event is generated by the security manager to the application when a passkey is required for pairing. When this event is received, the application has to respond with the ACI\_GAP\_PASS\_KEY\_RESP command.

**Table 411. ACI\_GAP\_PASS\_KEY\_REQ\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle for which the passkey has been requested	-

### 3.3.4 ACI\_GAP\_AUTHORIZATION\_REQ\_EVENT

This event is generated by the security manager when the application has set that an authorization is required for reading/writing of attributes. The event is generated as soon as the pairing is complete. When this event is received, an ACI\_GAP\_AUTHORIZATION\_RESP command must be used by the application to respond.

**Table 412. ACI\_GAP\_AUTHORIZATION\_REQ\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle for which authorization has been requested	-

### 3.3.5 ACI\_GAP\_PERIPHERAL\_SECURITY\_INITIATED\_EVENT

This event is generated when the peripheral security request is successfully sent to the central.

**Event parameters:** none

### 3.3.6 ACI\_GAP\_BOND\_LOST\_EVENT

This event is generated when a pairing request is issued in response to a peripheral security request from a central previously bonded with the peripheral. When this event is received, the upper layer must issue the command ACI\_GAP\_ALLOW\_REBOND to allow the peripheral to continue the pairing process with the central.

**Event parameters:** none

### 3.3.7 ACI\_GAP\_PROC\_COMPLETE\_EVENT

This event is sent by the GAP to the upper layers when a procedure previously started has been terminated by the upper layer, or has completed for any other reason.

**Table 413. ACI\_GAP\_PROC\_COMPLETE\_EVENT parameters**

Parameter	Size	Description	Possible values
Procedure_Code	1	Terminated procedure.	<ul style="list-style-type: none"><li>• 0x01: GAP_LIMITED_DISCOVERY_PROC</li><li>• 0x02: GAP_GENERAL_DISCOVERY_PROC</li><li>• 0x04: GAP_NAME_DISCOVERY_PROC</li><li>• 0x08: GAP_AUTO_CONNECTION_ESTABLISHMENT_PROC</li><li>• 0x10: GAP_GENERAL_CONNECTION_ESTABLISHMENT_PROC</li><li>• 0x20: GAP_SELECTIVE_CONNECTION_ESTABLISHMENT_PROC</li><li>• 0x40: GAP_DIRECT_CONNECTION_ESTABLISHMENT_PROC</li><li>• 0x80: GAP_OBSERVATION_PROC</li></ul>
Status	1	Error code	-
Data_Length	1	Length of data in octets	-

Parameter	Size	Description	Possible values
Data	Data_Length	Procedure specific data. For name discovery Procedure: the name of the peer device if the procedure completed successfully.	-

### 3.3.8 ACI\_GAP\_ADDR\_NOT\_RESOLVED\_EVENT

This event is sent only by a privacy enabled peripheral with a non empty bonded device list, when the peripheral is unsuccessful in resolving the resolvable address of the peer device after connecting to it.

**Table 414. ACI\_GAP\_ADDR\_NOT\_RESOLVED\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle for which the private address has not been resolved with any of the stored IRKs	-

### 3.3.9 ACI\_GAP\_NUMERIC\_COMPARISON\_VALUE\_EVENT

This event is sent only during SC v.4.2 pairing, when numeric comparison association model is selected, to show the numeric value generated, and to ask for confirmation to the user. When this event is received, the application must respond with the ACI\_GAP\_NUMERIC\_COMPARISON RESP command.

**Table 415. ACI\_GAP\_NUMERIC\_COMPARISON\_VALUE\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle related to the underlying pairing	-
Numeric_Value	4	-	-

### 3.3.10 ACI\_GAP\_KEYPRESS\_NOTIFICATION\_EVENT

This event is sent only during SC v.4.2 pairing, when keypress notifications are supported, to show the input type signaled by the peer device, having keyboard only I/O capabilities. When this event is received, no action is required to the user.

**Table 416. ACI\_GAP\_KEYPRESS\_NOTIFICATION\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle related to the underlying pairing	-
Notification_Type	1	Type of keypress input notified/signalized by peer device (having keyboard only I/O capabilities)	-

## 3.4

## ACI GATT/ATT events

In Table 417 "Y" means that the corresponding command applies to the dedicated BLE stack variant, namely LO / BO / PO / BF / LB, respectively, for Link layer only / Beacon only / Peripheral only / Basic feature / Link layer only basic.

Table 417. ACI GATT/ATT events list

Command	OpCode	LO	PO	BO	BF	LB
ACI_GATT_ATTRIBUTE_MODIFIED_EVENT	0x0C01	-	Y	-	Y	-
ACI_GATT_PROC_TIMEOUT_EVENT	0x0C02	-	Y	-	Y	-
ACI_ATT_EXCHANGE_MTU_RESP_EVENT	0x0C03	-	Y	-	Y	-
ACI_ATT_FIND_INFO_RESP_EVENT	0x0C04	-	-	-	Y	-
ACI_ATT_FIND_BY_TYPE_VALUE_RESP_EVENT	0x0C05	-	-	-	Y	-
ACI_ATT_READ_BY_TYPE_RESP_EVENT	0x0C06	-	-	-	Y	-
ACI_ATT_READ_RESP_EVENT	0x0C07	-	-	-	Y	-
ACI_ATT_READ_BLOB_RESP_EVENT	0x0C08	-	-	-	Y	-
ACI_ATT_READ_MULTIPLE_RESP_EVENT	0x0C09	-	-	-	Y	-
ACI_ATT_READ_BY_GROUP_TYPE_RESP_EVENT	0x0C0A	-	-	-	Y	-
ACI_ATT_PREPARE_WRITE_RESP_EVENT	0x0C0C	-	-	-	Y	-
ACI_ATT_EXEC_WRITE_RESP_EVENT	0x0C0D	-	-	-	Y	-
ACI_GATT_INDICATION_EVENT	0x0C0E	-	-	-	Y	-
ACI_GATT_NOTIFICATION_EVENT	0x0C0F	-	-	-	Y	-
ACI_GATT_PROC_COMPLETE_EVENT	0x0C10	-	-	-	Y	-
ACI_GATT_ERROR_RESP_EVENT	0x0C11	-	-	-	Y	-
ACI_GATT_DISC_READ_CHAR_BY_UUID_RESP_EVENT	0x0C12	-	-	-	Y	-
ACI_GATT_WRITE_PERMIT_REQ_EVENT	0x0C13	-	Y	-	Y	-
ACI_GATT_READ_PERMIT_REQ_EVENT	0x0C14	-	Y	-	Y	-
ACI_GATT_READ_MULTI_PERMIT_REQ_EVENT	0x0C15	-	Y	-	Y	-
ACI_GATT_TX_POOL_AVAILABLE_EVENT	0x0C16	-	Y	-	Y	-
ACI_GATT_SERVER_CONFIRMATION_EVENT	0x0C17	-	Y	-	Y	-
ACI_GATT_PREPARE_WRITE_PERMIT_REQ_EVENT	0x0C18	-	Y	-	Y	-
ACI_GATT_EATT_BEARER_EVENT	0x0C19	-	-	-	-	-
ACI_GATT_MULT_NOTIFICATION_EVENT	0xC1A	-	-	-	-	-
ACI_GATT_NOTIFICATION_COMPLETE_EVENT	0xC1B	-	Y	-	Y	-
ACI_GATT_READ_EXT_EVENT	0xC1D	-	-	-	Y	-
ACI_GATT_INDICATION_EXT_EVENT	0xC1E	-	-	-	Y	-
ACI_GATT_NOTIFICATION_EXT_EVENT	0xC1F	-	-	-	Y	-

### 3.4.1 ACI\_GATT\_ATTRIBUTE\_MODIFIED\_EVENT

This event is generated by the GATT server when a client modifies any attribute on the server, as consequence of one of the following GATT procedures:

- Write without response
- Signed write without response
- Write characteristic value
- Write long characteristic value - reliable write

**Table 418.** ACI\_GATT\_ATTRIBUTE\_MODIFIED\_EVENT parameters

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"><li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li><li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li></ul>
Attr_Handle	2	Handle of the modified attribute	-
Offset	2	Bits 14-0: offset from which the write has been performed by the peer device. Bit15 is used as flag: when set to 1 indicates that more data are to come (fragmented event in case of long attribute data).	-
Attr_Data_Length	2	Length of Attr_Data in octets	-
Attr_Data	Attr_Data_Length	Modified value	-

### 3.4.2 ACI\_GATT\_PROC\_TIMEOUT\_EVENT

This event is generated by the client/server to the application on a GATT timeout (30 seconds). It is a critical event that must not happen during normal operating conditions: it indicates either a major disruption in the communication link, or a mistake in the application, which does not provide a reply to GATT procedures. After the event, the GATT channel is closed and no more GATT communication can be performed. The application is expected to issue an ACI\_GAP\_TERMINATE to disconnect from the peer device. It is important to leave a 100 ms blank window before sending the ACI\_GAP\_TERMINATE, as, immediately after this event, the system could be saving important information in the non volatile memory.

**Table 419.** ACI\_GATT\_PROC\_TIMEOUT\_EVENT parameters

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"><li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li><li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li></ul>

### 3.4.3 ACI\_ATT\_EXCHANGE\_MTU\_RESP\_EVENT

This event is generated in response to an Exchange MTU request. See ACI\_GATT\_EXCHANGE\_CONFIG.

**Table 420.** ACI\_ATT\_EXCHANGE\_MTU\_RESP\_EVENT parameters

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle related to the response	0x0000 ... 0x0EFF
Server_RX_MTU	2	Attribute server receive MTU size	-

### 3.4.4 ACI\_ATT\_FIND\_INFO\_RESP\_EVENT

This event is generated in response to a find information request, or to ACI\_GATT\_DISC\_ALL\_CHAR\_DESC.

**Table 421. ACI\_ATT\_FIND\_INFO\_RESP\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"><li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li><li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li></ul>
Format	1	Format of Handle_UUID_Pair	-
Event_Data_Length	1	Length of Handle_UUID_Pair in octets	-
Handle_UUID_Pair	Event_Data_Length	A sequence of Handle-UUID pairs: <ul style="list-style-type: none"><li>• Format = 1: each pair is [2 octets for handle, 2 octets for UUIDs]</li><li>• Format = 2: each pair is [2 octets for handle, 16 octets for UUIDs]</li></ul>	-

### 3.4.5 ACI\_ATT\_FIND\_BY\_TYPE\_VALUE\_RESP\_EVENT

This event is generated in response to an ACI\_ATT\_FIND\_BY\_TYPE\_VALUE\_REQ.

**Table 422. ACI\_ATT\_FIND\_BY\_TYPE\_VALUE\_RESP\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"><li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li><li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li></ul>
Num_of_Handle_Pair	1	Number of attribute, group handle pairs	-
Found_Attribute_Handle[i]	2	Found attribute handle	-
Group_End_Handle[i]	2	Group end handle	-

### 3.4.6 ACI\_ATT\_READ\_BY\_TYPE\_RESP\_EVENT

This event is generated in response to an ACI\_ATT\_READ\_BY\_TYPE\_REQ. See ACI\_GATT\_FIND\_INCLUDED\_SERVICES and ACI\_GATT\_DISC\_ALL\_CHAR\_DESC.

**Table 423. ACI\_ATT\_READ\_BY\_TYPE\_RESP\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"> <li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li> <li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li> </ul>
Handle_Value_Pair_Length	1	The size of each attribute handle-value pair	-
Data_Length	1	Length of Handle_Value_Pair_Data, in octets	-
Handle_Value_Pair_Data	Data_Length	Attribute data. A sequence of handle-value pairs: [2 octets for Attribute Handle, (Handle_Value_Pair_Length - 2 octets) for attribute Value].	-

### 3.4.7 ACI\_ATT\_READ\_RESP\_EVENT

This event is generated in response to a read request. See [Section 2.5.24 ACI\\_GATT\\_READ\\_CHAR\\_VALUE](#).

**Table 424. ACI\_ATT\_READ\_RESP\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"> <li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li> <li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li> </ul>
Event_Data_Length	1	Length of following data	-
Attribute_Value	Event_Data_Length	Value of the attribute	-

### 3.4.8 ACI\_ATT\_READ\_BLOB\_RESP\_EVENT

This event can be generated during a read long characteristic value procedure, see [Section 2.5.26 ACI\\_GATT\\_READ\\_LONG\\_CHAR\\_VALUE](#).

**Table 425. ACI\_ATT\_READ\_BLOB\_RESP\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"> <li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li> <li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li> </ul>
Event_Data_Length	1	Length of following data	-
Attribute_Value	Event_Data_Length	Part of the attribute value	-

### 3.4.9 ACI\_ATT\_READ\_MULTIPLE\_RESP\_EVENT

This event is generated in response to a read multiple request.

**Table 426. ACI\_ATT\_READ\_MULTIPLE\_RESP\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"> <li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li> <li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li> </ul>
Event_Data_Length	1	Length of following data	-
Set_Of_Values	Event_Data_Length	A set of two or more values. A concatenation of attribute values for each attribute handles the request according to the order they were requested.	-

### 3.4.10 ACI\_ATT\_READ\_BY\_GROUP\_TYPE\_RESP\_EVENT

This event is generated in response to a read by group type request. See [Section 2.5.18 ACI\\_GATT\\_DISC\\_ALL\\_PRIMARY\\_SERVICES](#).

**Table 427. ACI\_ATT\_READ\_BY\_GROUP\_TYPE\_RESP\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"> <li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li> <li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li> </ul>
Attribute_Data_Length	1	Size of each attribute data	-
Data_Length	1	Length of Attribute_Data_List, in octets	-
Attribute_Data_List	Data_Length	Attribute data list: <ul style="list-style-type: none"> <li>• Attribute handle (2 octets)</li> <li>• End group handle (2 octets)</li> <li>• Attribute value (Attribute_Data_Length - 4 octets)</li> </ul>	-

### 3.4.11 ACI\_ATT\_PREPARE\_WRITE\_RESP\_EVENT

This event is generated in response to an ACI\_ATT\_PREPARE\_WRITE\_REQ.

**Table 428. ACI\_ATT\_PREPARE\_WRITE RESP\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"> <li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li> <li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li> </ul>
Attribute_Handle	2	Handle of the attribute to be written	-
Offset	2	Offset of the first octet to be written.	-
Part_Attribute_Value_Length	1	Length of Part_Attribute_Value in octets	-
Part_Attribute_Value	Part_Attribute_Value_Length	Value of the attribute to be written	-

### 3.4.12 ACI\_ATT\_EXEC\_WRITE\_RESP\_EVENT

This event is generated in response to an Execute Write Request.

**Table 429. ACI\_ATT\_EXEC\_WRITE RESP\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"> <li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li> <li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li> </ul>

### 3.4.13 ACI\_GATT\_INDICATION\_EVENT

This event is generated when an indication is received from the server.

**Table 430. ACI\_GATT\_INDICATION\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"> <li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li> <li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection oriented channel index)</li> </ul>
Attribute_Handle	2	Handle of the attribute	-
Attribute_Value_Length	1	Length of Attribute_Value in octets	-
Attribute_Value	Attribute_Value_Length	Current value of the attribute	-

### 3.4.14 ACI\_GATT\_NOTIFICATION\_EVENT

This event is generated when a notification is received from the server.

**Table 431. ACI\_GATT\_NOTIFICATION\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"><li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li><li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection oriented channel index)</li></ul>
Attribute_Handle	2	Handle of the attribute	-
Attribute_Value_Length	1	Length of Attribute_Value in octets	-
Attribute_Value	Attribute_Value_Length	Current value of the attribute	-

### 3.4.15 ACI\_GATT\_PROC\_COMPLETE\_EVENT

This event is generated when a GATT client procedure completes, with error or successfully.

**Table 432. ACI\_GATT\_PROC\_COMPLETE\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"><li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li><li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li></ul>
Error_Code	1	Indicates whether the procedure completed with an error or was successful (see <a href="#">Section 4 Status error codes</a> )	-

### 3.4.16 ACI\_GATT\_ERROR\_RESP\_EVENT

This event is generated when an error response is received from the server at the end of one of the GATT discovery procedures. This does not mean that the procedure ended with an error, but this error event is part of the procedure itself.

**Table 433. ACI\_GATT\_ERROR\_RESP\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"> <li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li> <li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li> </ul>
Req_Opcode	1	The request that generated the error response	-
Attribute_Handle	2	The attribute handle that generated the error response	-
Error_Code	1	The reason why the request has generated an error response (ATT error codes)	<ul style="list-style-type: none"> <li>• 0x01: Invalid handle</li> <li>• 0x02: Read not permitted</li> <li>• 0x03: Write not permitted</li> <li>• 0x04: Invalid PDU</li> <li>• 0x05: Insufficient authentication</li> <li>• 0x06: Request not supported</li> <li>• 0x07: Invalid offset</li> <li>• 0x08: Insufficient authorization</li> <li>• 0x09: Prepare queue full</li> <li>• 0x0A: Attribute not found</li> <li>• 0x0B: Attribute not long</li> <li>• 0x0C: Insufficient encryption key size</li> <li>• 0x0D: Invalid attribute value length</li> <li>• 0x0E: Unlikely error</li> <li>• 0x0F: Insufficient encryption</li> <li>• 0x10: Unsupported group type</li> <li>• 0x11: Insufficient resources</li> <li>• 0x12: Database out of synchronization</li> <li>• 0x13: Not allowed value</li> </ul>

### 3.4.17 ACI\_GATT\_DISC\_READ\_CHAR\_BY\_UUID\_RESP\_EVENT

This event can be generated during a "Discover Characteristics By UUID" or a "Read using Characteristic UUID" procedure. The attribute value is a service declaration, when a "Discover characteristics by UUID" has been started. It is the value of the characteristic if a "Read using characteristic UUID" has been performed.

**Table 434. ACI\_GATT\_DISC\_READ\_CHAR\_BY\_UUID\_RESP\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"> <li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li> <li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li> </ul>
Attribute_Handle	2	Handle of the attribute	-
Attribute_Value_Length	1	Length of Attribute_Value in octets	-
Attribute_Value	Attribute_Value_Length	The attribute value is a service, when a "Discover characteristics by UUID" has been started. It is the value of the characteristic if a "Read using characteristic UUID" has been performed.	-

### 3.4.18 ACI\_GATT\_WRITE\_PERMIT\_REQ\_EVENT

This event is given to the application when a write request, write command, or signed write command is received by the server. This happens only if the event bit for the generation is set when the characteristic was added.

When the event is received, the application must check whether the value requested for write can be written, and respond with the command ACI\_GATT\_WRITE\_RESP.

Based on the response from the application, the attribute value is modified by the stack. If the write is rejected by the application, the value of the attribute is not modified. In case of a write REQ, an error response is sent to the client, with the error code specified by the application. In case of write/signed write commands, no response is sent to the client, but the attribute is not modified.

**Table 435. ACI\_GATT\_WRITE\_PERMIT\_REQ\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"><li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li><li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li></ul>
Attribute_Handle	2	Handle of the attribute	-
Data_Length	1	Length of data field	-
Data	Data_Length	The data that the client has requested to write	-

### 3.4.19 ACI\_GATT\_READ\_PERMIT\_REQ\_EVENT

This event is given to the application when a read or a read blob request is received by the server from the client, only if the event bit for the generation is set when the characteristic was added. Upon the reception of the event, the application can update the value of the handle, and when done, it must send the ACI\_GATT\_ALLOW\_READ command to indicate to the stack that it can send the response to the client.

**Table 436. ACI\_GATT\_READ\_PERMIT\_REQ\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"><li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li><li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li></ul>
Attribute_Handle	2	The handle of the attribute	-
Offset	2	Contains the offset from which the read has been requested	-

### 3.4.20 ACI\_GATT\_READ\_MULTI\_PERMIT\_REQ\_EVENT

This event is given to the application when a read multiple request or read by type request is received by the server. This happens only if the event bit for the generation is set when the characteristic was added. On the reception of this event, the application can update the value of the handles (if needed), and, when done, must send the ACI\_GATT\_ALLOW\_READ command to indicate to the stack that it can send the response to the client.

**Table 437. ACI\_GATT\_READ\_MULTI\_PERMIT\_REQ\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"><li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li><li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li></ul>
Number_of_Handles	1	-	-
Handle[i]	2	-	-

### 3.4.21 ACI\_GATT\_TX\_POOL\_AVAILABLE\_EVENT

This event is generated each time BLE FW stack raises the error code BLE\_STATUS\_INSUFFICIENT\_RESOURCES (0x64), as soon as there are at least two buffers available for notifications or write commands.

**Table 438. ACI\_GATT\_TX\_POOL\_AVAILABLE\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Not used	-
Available_Buffers	2	Number of available buffers	-

### 3.4.22 ACI\_GATT\_SERVER\_CONFIRMATION\_EVENT

This event is generated when the client has sent the confirmation for a previously sent indication.

**Table 439. ACI\_GATT\_SERVER\_CONFIRMATION\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"><li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li><li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li></ul>

### 3.4.23 ACI\_GATT\_PREPARE\_WRITE\_PERMIT\_REQ\_EVENT

This event is given to the application when a Prepare write request is received by the server from the client (if the event bit for the generation is set when the characteristic was added). When the event is received, the application must check whether the value requested for write can be written, and respond with the command ACI\_GATT\_WRITE\_RESP. Based on the response from the application, the attribute value is modified by the stack. If the write is rejected, the value of the attribute is not modified, and an error response is sent to the client, with the error code specified by the application.

**Table 440. ACI\_GATT\_PREPARE\_WRITE\_PERMIT\_REQ\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"> <li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li> <li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li> </ul>
Attribute_Handle	2	Handle of the attribute	-
Offset	2	Offset from which the Prepare write has been requested	-
Data_Length	1	Length of Data field	-
Data	Data_Length	The data that the client has requested to write	-

**3.4.24****ACI\_GATT\_EATT\_BEARER\_EVENT**

This event informs the application of a change in status of the Enhanced ATT bearer handled by the specified L2CAP channel.

**Table 441. ACI\_GATT\_EATT\_BEARER\_EVENT parameters**

Parameter	Size	Description	Possible values
Channel_Index	1	Index of the connection-oriented channel to whom the primitive applies	-
EAB_State	1	Enhanced ATT bearer state	<ul style="list-style-type: none"> <li>• 0x00: enhanced ATT bearer created</li> <li>• 0x01: enhanced ATT bearer terminated</li> </ul>
Status	1	Status error code	-

**3.4.25****ACI\_GATT\_MULT\_NOTIFICATION\_EVENT**

This event is generated when a Miultiple\_Handle\_Value notification is received from the server.

**Table 442. ACI\_GATT\_MULT\_NOTIFICATION\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Index of the connection-oriented channel to whom the primitive applies	<ul style="list-style-type: none"> <li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li> <li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li> </ul>
Offset	2	<ul style="list-style-type: none"> <li>• Bits 14-0: offset in octets from which Attribute_Value data starts</li> <li>• Bit 15: flag, when set to 1 it indicates that more data are to come (fragmented event in case of long attribute data)</li> </ul>	-
Data_Length	2	Length of Data, in bytes	-
Data	Data_Length	List of "Handle Length Value" tuples, as defined in Bluetooth Core specification	-

### 3.4.26 ACI\_GATT\_NOTIFICATION\_COMPLETE\_EVENT

This event is generated on the server side after the transmission of all notifications linked with a local update of a characteristic value (if enabled at the creation of the characteristic with GATT\_NOTIFY\_NOTIFICATION\_COMPLETION mask and if the characteristic supports notifications).

**Table 443. ACI\_GATT\_NOTIFICATION\_COMPLETE\_EVENT parameters**

Parameter	Size	Description	Possible values
Attr_Handle	2	Handle of the updated characteristic value	-

### 3.4.27 ACI\_GATT\_READ\_EXT\_EVENT

When enabled with ACI\_GATT\_SET\_EVENT\_MASK, this event is generated instead of ACI\_ATT\_READ\_RESP\_EVENT / ACI\_ATT\_READ\_BLOB\_RESP\_EVENT / ACI\_ATT\_READ\_MULTIPLE\_RESP\_EVENT. To be used when ATT\_MTU > (BLE\_EVT\_MAX\_PARAM\_LEN - 4), hence ATT\_MTU > 251 for the BLE\_EVT\_MAX\_PARAM\_LEN default value.

**Table 444. ACI\_GATT\_READ\_EXT\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"><li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li><li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li></ul>
Offset	2	Bits 14-0: offset from which Attribute_Value data starts, in octets Bit 15 is used as flag: when set to 1 it indicates that more data are to come (fragmented event in case of long attribute data)	-
Event_Data_Length	2	Length of following data	-
Attribute_Value	Event_Data_Length	The value of the attribute(s)	-

### 3.4.28 ACI\_GATT\_INDICATION\_EXT\_EVENT

When enabled with ACI\_GATT\_SET\_EVENT\_MASK, and when an indication is received from the server, this event is generated instead of ACI\_GATT\_INDICATION\_EVENT. This event is used instead of ACI\_GATT\_INDICATION\_EVENT when ATT\_MTU > (BLE\_EVT\_MAX\_PARAM\_LEN - 4), hence when ATT\_MTU > 251 for the BLE\_EVT\_MAX\_PARAM\_LEN default value.

**Table 445. ACI\_GATT\_INDICATION\_EXT\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"> <li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li> <li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li> </ul>
Attribute_Handle	2	The handle of the attribute	-
Offset	2	Bits 14-0: offset from which Attribute_Value data starts, in octets  Bit 15 is used as flag: when set to 1 it indicates that more data are to come (fragmented event in case of long attribute data).	-
Data_Length	2	Length of Attribute_Value in octets	-
Data	Attribute_Value_Length	The current value of the attribute	-

### 3.4.29 ACI\_GATT\_NOTIFICATION\_EXT\_EVENT

This event is generated instead of ACI\_GATT\_NOTIFICATION\_EVENT when enabled with ACI\_GATT\_SET\_EVENT\_MASK, and when a notification is received from the server. It is used instead of ACI\_GATT\_NOTIFICATION\_EVENT when ATT\_MTU > (BLE\_EVT\_MAX\_PARAM\_LEN - 4), as an example, when ATT\_MTU > 251 for the BLE\_EVT\_MAX\_PARAM\_LEN default value.

**Table 446. ACI\_GATT\_NOTIFICATION\_EXT\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Specifies the ATT bearer to whom the command applies	<ul style="list-style-type: none"> <li>• 0x0000 ... 0x0EFF: unenhanced ATT bearer (the parameter is the connection handle)</li> <li>• 0xEA00 ... 0xEA3F: enhanced ATT bearer (the parameter LSB is the connection-oriented channel index)</li> </ul>
Attribute_Handle	2	Handle of the attribute	-
Offset	2	Bits 14-0: offset in octets from which Attribute_Value data starts.  Bit 15 is used as flag: when set to 1 it indicates that more data are to come (fragmented event in case of long attribute data).	-
Data_Length		Length of Attribute_Value in octets	-
Data	Attribute_Value_Length	Current value of the attribute	-

### 3.5

## ACI L2CAP events

In Table 447 "Y" means that the corresponding command applies to the dedicated BLE stack variant, namely LO / BO / PO / BF / LB, respectively, for Link layer only / Beacon only / Peripheral only / Basic feature / Link layer only basic.

**Table 447. ACI L2CAP events commands list**

Command	OpCode	LO	PO	BO	BF	LB
ACI_L2CAP_CONNECTION_UPDATE_RESP_EVENT	0x0800	-	Y	-	Y	-
ACI_L2CAP_PROC_TIMEOUT_EVENT	0x0801	-	Y	-	Y	-
ACI_L2CAP_CONNECTION_UPDATE_REQ_EVENT	0x0802	-	-	-	Y	-
ACI_L2CAP_COMMAND_REJECT_EVENT	0x080A	-	Y	-	Y	-
ACI_L2CAP_CO_C_CONNECT_EVENT	0x0810	-	-	-	-	-
ACI_L2CAP_CO_C_CONNECT_CONFIRM_EVENT	0x0811	-	-	-	-	-
ACI_L2CAP_CO_C_RECONF_EVENT	0x0812	-	-	-	-	-
ACI_L2CAP_CO_C_RECONF_CONFIRM_EVENT	0x0813	-	-	-	-	-
ACI_L2CAP_CO_C_DISCONNECT_EVENT	0x0814	-	-	-	-	-
ACI_L2CAP_CO_C_FLOW_CONTROL_EVENT	0x0815	-	-	-	-	-
ACI_L2CAP_CO_C_RX_DATA_EVENT	0x0816	-	-	-	-	-
ACI_L2CAP_CO_C_TX_POOL_AVAILABLE_EVENT	0x0817	-	-	-	-	-

### 3.5.1

#### ACI\_L2CAP\_CONNECTION\_UPDATE\_RESP\_EVENT

This event is generated when the central responds to the connection update request packet with a connection update response packet.

**Table 448. ACI\_L2CAP\_CONNECTION\_UPDATE\_RESP\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle referring to the COS channel where the disconnection has been received	-
Result	2	-	-

### 3.5.2

#### ACI\_L2CAP\_PROC\_TIMEOUT\_EVENT

This event is generated when the central does not respond to the connection update request packet with a connection update response packet or a command reject packet within 30 seconds.

**Table 449. ACI\_L2CAP\_PROC\_TIMEOUT\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Handle of the connection related to this L2CAP procedure	-
Data_Length	1	Length of following data	-
Data	Data_Length	-	-

### 3.5.3

#### ACI\_L2CAP\_CONNECTION\_UPDATE\_REQ\_EVENT

The event is given by the L2CAP layer when a connection update request is received from the peripheral. The upper layer receiving this event must respond by sending an ACI\_L2CAP\_CONNECTION\_PARAMETER\_UPDATE\_RESP command.

**Table 450. ACI\_L2CAP\_CONNECTION\_UPDATE\_REQ\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Handle of the connection related to this L2CAP procedure	-
Identifier	1	This is the identifier which associate the request to the response	-
L2CAP_Length	2	Length of the L2CAP connection update request	-
Interval_Min	2	Minimum value for the connection event interval. This is less than or equal to Conn_Interval_Max.  Time = N * 1.25 ms	0x0006 (7.50 ms) ... 0x0C80 (4000.00 ms)
Interval_Max	2	Maximum value for the connection event interval. This is greater than or equal to Conn_Interval_Min.  Time = N * 1.25 ms	0x0006 (7.50 ms) ... 0x0C80 (4000.00 ms)
Conn_Latency	2	Maximum peripheral latency for the connection, in number of connection events	0x0000 ... 0x01F3
Timeout_Multiplier	2	Defines connection timeout parameter as: Timeout Multiplier * 10 ms.	-

### 3.5.4 ACI\_L2CAP\_COMMAND\_REJECT\_EVENT

This event is generated when the central responds to the connection update request packet with a command reject packet.

**Table 451. ACI\_L2CAP\_COMMAND\_REJECT\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Connection handle referring to the COS channel where the disconnection has been received	-
Identifier	1	Identifier that associates the request to the response	-
Reason	2	Reason	-
Data_Length	1	Length of following data	-
Data	Data_Length	Data field associated with reason	-

### 3.5.5 ACI\_L2CAP\_CO\_C\_CONNECT\_EVENT

This event is generated when receiving a valid credit based connection request packet.

**Table 452. ACI\_L2CAP\_CO\_C\_CONNECT\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Handle of the connection where this event occurred	0x0000 ... 0x0EFF
SMPS	2	Simplified Protocol/Service Multiplexer	0x0000 ... 0x00FF
MTU	2	Maximum transmission unit	23 ... 65535
MPS	2	Maximum payload size, in octects	23 ... 248
Initial_Credits	2	Number of K-frames that can be received on the created channel(s) by the L2CAP layer entity sending this packet	0 ... 65535
Channel_Number	1	Number of channels to be created. If set to 0, this parameter requests the creation of one LE credit based connection-oriented channel, otherwise it requests the creation of one or more enhanced credit based connection-oriented channels.	0 ... 5

### 3.5.6 ACI\_L2CAP\_CO\_C\_CONNECT\_CONFIRM\_EVENT

This event is generated when receiving a valid credit based connection response packet. See Bluetooth spec. v.5.2 [Vol 3, Part A].

**Table 453. ACI\_L2CAP\_CO\_C\_CONNECT\_CONFIRM\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Handle of the connection where this event occurred.	0x0000 ... 0x0EFF
MTU	2	Maximum transmission unit	23 ... 65535
MPS	2	Maximum payload size, in octects	23 ... 248
Initial_Credits	2	Number of K-frames that can be received on the created channel(s) by the L2CAP layer entity sending this packet.	0 ... 65535
Results	2	This parameter indicates the outcome of the request. A value of 0x0000 indicates success, a non-zero value indicates the request is refused.	0x0000 ... 0x000C
Channel_Number	1	Number of created channels. It is the length of Channel_Index_List.	0 ... 5
Channel_Index_List	Channel_Number	List of channel indexes to whom the primitive applies	-

### 3.5.7 ACI\_L2CAP\_CO\_C\_RECONF\_EVENT

This event is generated when receiving a valid credit based reconfigure request packet. See Bluetooth spec. v.5.2 [Vol 3, Part A].

**Table 454. ACI\_L2CAP\_CO\_C\_RECONF\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Handle of the connection where this event occurred	0x0000 ... 0x0EFF
MTU	2	Maximum transmission unit	23 ... 65535
MPS	2	Maximum payload size, in octects	23 ... 248
Channel_Number	1	Number of created channels. It is the length of Channel_Index_List.	1 ... 5
Channel_Index_List	Channel_Number	List of channel indexes to whom the primitive applies	-

### 3.5.8 ACI\_L2CAP\_CO\_C\_RECONF\_CONFIRM\_EVENT

This event is generated when receiving a valid credit based reconfigure response packet. See Bluetooth spec. v.5.2 [Vol 3, Part A].

**Table 455. ACI\_L2CAP\_CO\_C\_RECONF\_CONFIRM\_EVENT parameters**

Parameter	Size	Description	Possible values
Connection_Handle	2	Handle of the connection where this event occurred.	0x0000 ... 0x0EFF
Result	2	This parameter indicates the outcome of the request. A value of 0x0000 indicates success, a non-zero value indicates that the request is refused.	0x0000 ... 0x000C

### 3.5.9 ACI\_L2CAP\_CO\_C\_DISCONNECT\_EVENT

This event is generated when a connection-oriented channel is disconnected following an L2CAP channel termination procedure. See Bluetooth spec. v.5.2 [Vol 3, Part A].

**Table 456. ACI\_L2CAP\_CO\_C\_DISCONNECT\_EVENT parameters**

Parameter	Size	Description	Possible values
Channel_Index	1	Index of the connection-oriented channel to whom the primitive applies.	-

### 3.5.10 ACI\_L2CAP\_CO\_C\_FLOW\_CONTROL\_EVENT

This event is generated when receiving a valid flow control credit signaling packet. See Bluetooth spec. v.5.2 [Vol 3, Part A].

**Table 457. ACI\_L2CAP\_CO\_C\_DISCONNECT\_EVENT parameters**

Parameter	Size	Description	Possible values
Channel_Index	1	Index of the connection-oriented channel to whom the primitive applies	-
Credits	2	Number of credits the receiving device can increment, corresponding to the number of K-frames that can be sent to the peer device sending the Flow Control Credit packet.	1 ... 65535

### 3.5.11 ACI\_L2CAP\_CO\_C\_RX\_DATA\_EVENT

This event is generated when receiving a valid K-frame packet on a connection-oriented channel. See Bluetooth spec. v.5.2 [Vol 3, Part A].

**Note:** *For the first K-frame of the SDU, the Information data contains the L2CAP SDU length coded on two octets, followed by the K-frame information payload. For the next K-frames of the SDU the information data only contains the K-frame information payload.*

**Table 458. ACI\_L2CAP\_CO\_C\_RX\_DATA\_EVENT parameters**

Parameter	Size	Description	Possible values
Channel_Index	1	Index of the connection-oriented channel to whom the primitive applies.	-
Length	2	Length of data in octects	-
Data	Length	Information data	-

### 3.5.12 ACI\_L2CAP\_CO\_C\_TX\_POOL\_AVAILABLE\_EVENT

Each time ACI\_L2CAP\_CO\_C\_TX\_DATA raises the error code BLE\_STATUS\_INSUFFICIENT\_RESOURCES (0x64), the ACI\_L2CAP\_CO\_C\_TX\_POOL\_AVAILABLE\_EVENT event is generated as soon as there is a free buffer available for sending K-frames.

**Event parameters:** none

## 3.6 ACI HAL events

In Table 459 "Y" means that the corresponding command applies to the dedicated BLE stack variant, namely LO / BO / PO / BF / LB, respectively, for Link layer only / Beacon only / Peripheral only / Basic feature / Link layer only basic.

Table 459. ACI HAL events commands list

Command	OpCode	LO	PO	BO	BF	LB
ACI_HAL_END_OF_RADIO_ACTIVITY_EVENT	0x0004	Y	Y	Y	Y	Y
ACI_HAL_SCAN_REQ_REPORT_EVENT	0x0005	-	-	-	-	-
ACI_HAL_FW_ERROR_EVENT	0x0006	-	Y	-	Y	-

### 3.6.1 ACI\_HAL\_END\_OF\_RADIO\_ACTIVITY\_EVENT

This event is generated when the device completes a radio activity, and provides information when a new radio activity is performed. The provided information includes the type of radio activity and the absolute time in system ticks when a new radio activity is scheduled, if any. The application uses this information to schedule user activities synchronous to selected radio activities. An ACI\_HAL\_SET\_RADIO\_ACTIVITY\_MASK command is provided to enable radio activity events of user interests (by default no events are enabled). The enabling radio events in applications with intense radio activity can lead to a fairly high rate of generated events. The application use cases include synchronizing notification with connection interval, switching antenna at the end of advertising, or performing flash erase operation while radio is idle.

Table 460. ACI\_HAL\_END\_OF\_RADIO\_ACTIVITY\_EVENT parameters

Parameter	Size	Description	Possible values
Last_State	1	Completed radio events	<ul style="list-style-type: none"><li>• 0x00: Idle</li><li>• 0x01: Advertising</li><li>• 0x02: Connection event peripheral</li><li>• 0x03: Scanning</li><li>• 0x04: Connection request</li><li>• 0x05: Connection event peripheral</li><li>• 0x06: TX test mode</li><li>• 0x07: RX test mode</li></ul>
Next_State	1	Incoming radio events	<ul style="list-style-type: none"><li>• 0x00: Idle</li><li>• 0x01: Advertising</li><li>• 0x02: Connection event peripheral</li><li>• 0x03: Scanning</li><li>• 0x04: Connection request</li><li>• 0x05: Connection event peripheral</li><li>• 0x06: TX test mode</li><li>• 0x07: RX test mode</li></ul>
Next_State_SysTime	4	32-bit absolute current time expressed in internal time units	-
Last_state_Slot	1	Slot number of completed radio event	<ul style="list-style-type: none"><li>• 0xFF: Idle</li><li>• 0x00 ... 0x07</li></ul>
Next_State_Slot	1	Slot number of incoming radio event	<ul style="list-style-type: none"><li>• 0xFF: Idle</li><li>• 0x00 ... 0x07</li></ul>

### 3.6.2 ACI\_HAL\_SCAN\_REQ\_REPORT\_EVENT

This event is reported to the application after a scan request is received and a scan reponse is scheduled for transmission.

**Table 461.** ACI\_HAL\_SCAN\_REQ\_REPORT\_EVENT parameters

Parameter	Size	Description	Possible values
RSSI	1	N Size: 1 octet (signed integer), in dBm	<ul style="list-style-type: none"><li>• 127: RSSI not available</li><li>• -127 ... 20</li></ul>
Peer_Address_Type	1	Peer address type	<ul style="list-style-type: none"><li>• 0x00: public device address</li><li>• 0x01: random device address</li><li>• 0x02: public identity address</li><li>• 0x03: random (static) identity address</li></ul>
Peer_Address	6	Public or random device address of the peer device	-

### 3.6.3 ACI\_HAL\_FW\_ERROR\_EVENT

This event is generated to report firmware error information.

**Table 462.** ACI\_HAL\_FW\_ERROR\_EVENT parameters

Parameter	Size	Description	Possible values
FW_Error_Type	1	FW error type	<ul style="list-style-type: none"><li>• 0x01: L2CAP recombination failure</li><li>• 0x02: GATT unexpected peer message</li><li>• 0x03: NVM level warning</li><li>• 0x04: COC RX data length too large</li><li>• 0x05: ECOC already assigned DCID</li></ul>
Data_Length	1	Length of data, in octets	-
Data	Data_Length	The error event info	-

## 4 Status error codes

Status error codes are used for the return status of all commands. The codes from 0x00 to 0x3E are used for HCI commands (see Core Specification v5.2, Vol. 2, part D), other codes are defined for ACI commands (see the following table).

Table 463. Status error codes description

Status error code	Description
0x00	Success
0x01	Unknown HCI command
0x02	Unknown connection identifier
0x03	Hardware failure
0x05	Authentication failure
0x06	PIN or key missing
0x07	Memory capacity exceeded
0x08	Connection timeout
0x09	Connection limit exceeded
0x0B	ACL connection already exists
0x0C	Command disallowed
0x11	Unsupported feature or parameter value
0x12	Invalid HCI command parameters
0x13	Remote user terminated connection
0x15	Remote device terminated connection due to power off
0x16	Connection terminated by local host
0x1A	Unsupported remote feature
0x1E	Invalid LL parameters
0x1F	Unspecified error
0x22	LL response timeout
0x23	LL procedure collision
0x24	LMP PDU not allowed
0x28	Instant passed
0x2A	Different transaction collision
0x2E	Channel assessment not supported
0x2F	Insufficient security
0x30	Parameter out of mandatory range
0x38	Host busy - pairing
0x3A	Controller busy
0x3B	Unacceptable connection interval
0x3C	Directed advertising timeout
0x3D	Connection terminated due to MIC failure
0x3E	Connection failed to be established
0x42	Unknown advertising identifier
0x43	Limit reached

Status error code	Description
0x45	Packet too long
0x59	Device in blacklist
0x5A	CSRK not found
0x5B	IRK not found
0x5C	Device not found in DB
0x5E	Device not bonded
0x60	Invalid handle
0x61	Out of handles
0x62	Invalid operation
0x64	Insufficient resources
0x65	Security permission error
0x70	Address not resolved
0x82	No valid slot
0x83	Short window
0x84	New interval failed
0x85	Too large interval
0x86	Slot length failed
0x91	Failed
0x92	Invalid parameters
0x93	Busy
0x95	Pending
0x97	Host error
0x98	Out of memory

## 5 Tx power level

The following table gives the output power corresponding to the possible values of PA\_Level parameter (see [Section 2.3.4 ACI\\_HAL\\_SET\\_TX\\_POWER\\_LEVEL](#)). These values (at the MCU output) are indicative, and depend upon the PCB layout and the associated components.

**Table 464. Tx power level**

PA_Level	Output power (in dBm)
0x00	-40.0
0x01	-20.85
0x02	-19.75
0x03	-18.85
0x04	-17.60
0x05	-16.50
0x06	-15.25
0x07	-14.10
0x08	-13.15
0x09	-12.05
0x0A	-10.90
0x0B	-9.90
0x0C	-8.85
0x0D	-7.80
0x0E	-6.90
0x0F	-5.90
0x10	-4.95
0x11	-4.00
0x12	-3.15
0x13	-2.45
0x14	-1.80
0x15	-1.30
0x16	-0.85
0x17	-0.50
0x18	-0.15
0x19	0
0x1A	+1.0
0x1B	+2.0
0x1C	+3.0
0x1D	+4.0
0x1E	+5.0
0x1F	+6.0

## Revision history

Table 465. Document revision history

Date	Version	Changes
20-Feb-2019	1	Initial release.
04-Jun-2019	2	Updated <a href="#">Table 6. HCI_SET_EVENT_MASK input parameters</a> , <a href="#">Table 388. HCI_HARDWARE_ERROR_EVENT parameters</a> , <a href="#">Table 462. ACI_HAL_FW_ERROR_EVENT parameters</a>
17-Jun-2019	3	<p>Updated:</p> <ul style="list-style-type: none"><li><a href="#">Table 280. ACI_GATT_UPDATE_CHAR_VALUE input parameters</a>,</li><li><a href="#">Table 288. ACI_GATT_SET_EVENT_MASK input parameters</a>,</li><li><a href="#">Table 316. ACI_GATT_READ_CHAR_VALUE input parameters</a>,</li><li><a href="#">Table 320. ACI_GATT_READ_LONG_CHAR_VALUE input parameters</a>,</li><li><a href="#">Table 324. ACI_GATT_WRITE_CHAR_VALUE input parameters</a>,</li><li><a href="#">Table 326. ACI_GATT_WRITE_LONG_CHAR_VALUE input parameters</a>,</li><li><a href="#">Table 338. ACI_GATT_WRITE_WITHOUT_RESP input parameters</a>,</li><li><a href="#">Table 340. ACI_GATT_SIGNED_WRITE_WITHOUT_RESP input parameters</a>,</li><li><a href="#">Table 354. ACI_GATT_UPDATE_CHAR_VALUE_EXT input parameters</a>, <a href="#">Table 417. ACI_GATT/ATT events list</a></li></ul> <p>Added:</p> <ul style="list-style-type: none"><li><a href="#">Section 3.4.28 ACI_GATT_INDICATION_EXT_EVENT</a> and <a href="#">Section 3.4.29 ACI_GATT_NOTIFICATION_EXT_EVENT</a></li></ul>
25-Sep-2019	4	<p>Updated:</p> <ul style="list-style-type: none"><li><a href="#">Section 2.4.18 ACI_GAP_CONFIGURE_FILTER_ACCEPT_LIST</a>,</li><li><a href="#">Section 2.5.14 ACI_ATT_READ_BY_TYPE_REQ</a>,</li><li><a href="#">Section 2.5.25 ACI_GATT_READ_USING_CHAR_UUID</a>,</li><li><a href="#">Section 3.4.28 ACI_GATT_INDICATION_EXT_EVENT</a>, <a href="#">Section 4 Status error codes</a></li><li><a href="#">Table 219. ACI_GAP_TERMINATE_GAP_PROC input parameters</a>,</li><li><a href="#">Table 288. ACI_GATT_SET_EVENT_MASK input parameters</a>,</li><li><a href="#">Table 385. HCI_DISCONNECTION_COMPLETE_EVENT parameters</a>,</li><li><a href="#">Table 413. ACI_GAP_PROC_COMPLETE_EVENT parameters</a>, <a href="#">Table 417. ACI GATT/ATT events list</a>, <a href="#">Table 418. ACI_GATT_ATTRIBUTE_MODIFIED_EVENT parameters</a>,</li><li><a href="#">Table 445. ACI_GATT_INDICATION_EXT_EVENT parameters</a>,</li><li><a href="#">Table 446. ACI_GATT_NOTIFICATION_EXT_EVENT parameters</a>, <a href="#">Table 463. Status error codes description</a></li></ul> <p>Added <a href="#">Section 3.4.27 ACI_GATT_READ_EXT_EVENT</a></p> <p>Removed former <a href="#">Section 4.2 ATT error codes</a></p>
07-Jan-2020	5	<p>Updated <a href="#">Table 138. ACI_HAL_WRITE_CONFIG_DATA input parameters</a>, <a href="#">Table 140. ACI_HAL_READ_CONFIG_DATA input parameters</a>, and <a href="#">Table 410. ACI_GAP_PAIRING_COMPLETE_EVENT parameters</a>.</p> <p>Updated <a href="#">Section 2.5.35 ACI_GATT_WRITE_WITHOUT_RESP</a> and <a href="#">Section 2.5.36 ACI_GATT_SIGNED_WRITE_WITHOUT_RESP</a>.</p>
26-May-2020	6	<p>Updated <a href="#">Table 6. HCI_SET_EVENT_MASK input parameters</a>, <a href="#">Table 138. ACI_HAL_WRITE_CONFIG_DATA input parameters</a>, <a href="#">Table 140. ACI_HAL_READ_CONFIG_DATA input parameters</a>, <a href="#">Table 385. HCI_DISCONNECTION_COMPLETE_EVENT parameters</a>, <a href="#">Table 410. ACI_GAP_PAIRING_COMPLETE_EVENT parameters</a>, <a href="#">Table 432. ACI_GATT_PROC_COMPLETE_EVENT parameters</a>, <a href="#">Table 462. ACI_HAL_FW_ERROR_EVENT parameters</a> and <a href="#">Table 463. Status error codes description</a>.</p> <p>Updated title of <a href="#">Section 2.1.34 HCI_LE_READ_REMOTE_FEATURES</a> and <a href="#">Section 3.2.4 HCI_LE_READ_REMOTE_FEATURES_COMPLETE_EVENT</a>.</p> <p>Updated <a href="#">Section 2.4.22 ACI_GAP_START_LIMITED_DISCOVERY_PROC</a> and <a href="#">Section 2.4.23 ACI_GAP_START_GENERAL_DISCOVERY_PROC</a>.</p> <p>Added note in <a href="#">Section 2.4.10 ACI_GAP_INIT</a>.</p> <p>Removed former section <a href="#">HCI_DATA_BUFFER_OVERFLOW_EVENT</a>.</p>
16-Jul-2020	7	Updated <a href="#">Section 3.2.2 HCI_LE_ADVERTISING_REPORT_EVENT</a> and removed former <a href="#">Section 3.6.4 ACI_HAL_DATAPUMP_SENT_EVENT</a> .

Date	Version	Changes
16-Jul-2020	7 (cont'd)	<p>Added three columns in Table 1. HCI commands list, Table 126. HCI testing commands list, Table 136. HAL commands list, Table 167. GAP commands list, Table 270. GATT/ATT commands list, Table 365. L2CAP commands list, Table 384. HCI events commands list, Table 393. HCI LE META events command list, Table 409. ACI GAP events commands list, Table 417. ACI GATT/ATT events list, Table 447. ACI L2CAP events commands list, and Table 459. ACI HAL events commands list.</p>
03-Nov-2020	8	<p>Updated document title, Introduction, Section 1 General information, Section 2.1 HCI commands, Section 2.2 HCI testing commands, Section 2.3 HAL commands, Section 2.4 GAP commands, Section 2.5 GATT/ATT commands, Section 2.6 L2CAP commands, and Section 3.1 HCI events.</p> <p>Updated Table 1. HCI commands list, Table 136. HAL commands list and Table 410. ACI_GAP_PAIRING_COMPLETE_EVENT parameters.</p> <p>Removed former Section 2.3.12 ACI_HAL_SET_SMP_ENG_CONFIG.</p> <p>Minor text edits across the whole document.</p>
21-Jan-2021	9	<p>Updated Introduction, Section 3.2 HCI LE META events, Section 3.3 ACI GAP events, and Section 4 Status error codes.</p> <p>Updated Table 148. ACI_HAL_GET_LINK_STATUS output parameters.</p> <p>Minor text edits across the whole document.</p>
18-Jun-2021	10	<p>Updated Section 2.1 HCI commands, Section 2.1.6 HCI_SET_CONTROLLER_TO_HOST_FLOW_CONTROL, Section 2.1.7 HCI_HOST_BUFFER_SIZE, Section 2.2 HCI testing commands, Section 2.3 HAL commands, Section 2.4 GAP commands, Section 2.5 GATT/ATT commands, Section 2.6 L2CAP commands, and Section 3.5 ACI L2CAP events.</p> <p>Updated Table 1. HCI commands list, Table 126. HCI testing commands list, Table 136. HAL commands list, Table 167. GAP commands list, Table 270. GATT/ATT commands list, Table 365. L2CAP commands list, and Table 447. ACI L2CAP events commands list.</p> <p>Added Section 2.1.70 HCI_LE_READ_TRANSMIT_POWER, Section 2.1.73 HCI_LE_SET_PRIVACY_MODE, Section 2.4.43 ACI_GAP_ADDITIONAL_BEACON_START, Section 2.4.44 ACI_GAP_ADDITIONAL_BEACON_STOP, Section 2.4.45 ACI_GAP_ADDITIONAL_BEACON_SET_DATA, Section 3.5.5 to Section 3.5.9, and Section 3.5.5 to Section 3.5.12.</p> <p>Minor text edits across the whole document.</p>
17-Dec-2021	11	<p>Updated Table 1. HCI commands list, Table 126. HCI testing commands list, Table 136. HAL commands list, Table 138. ACI_HAL_WRITE_CONFIG_DATA input parameters, Table 167. GAP commands list, Table 169. ACI_GAP_SET_LIMITED_DISCOVERABLE input parameters, Table 171. ACI_GAP_SET_DISCOVERABLE input parameters, Table 173. ACI_GAP_SET_DIRECT_CONNECTABLE input parameters, Table 185. ACI_GAP_INIT input parameters, Table 213. ACI_GAP_START_GENERAL_CONNECTION_ESTABLISH_PROC input parameters, Table 215. ACI_GAP_START_SELECTIVE_CONNECTION_ESTABLISH_PROC input parameters, Table 217. ACI_GAP_CREATE_CONNECTION input parameters, Table 227. ACI_GAP_SET_BROADCAST_MODE input parameters, Table 229. ACI_GAP_START_OBSERVATION_PROC input parameters, Table 239. ACI_GAP_GET_OOB_DATA output parameters, Table 240. ACI_GAP_SET_OOB_DATA input parameters, Table 242. ACI_GAP_ADD_DEVICES_TO_RESOLVING_LIST input parameters, Table 393. HCI LE META events command list, Table 417. ACI GATT/ATT events list, Table 462. ACI_HAL_FW_ERROR_EVENT parameters, and Table 463. Status error codes description.</p>

Date	Version	Changes
17-Dec-2021	11 (cont'd)	<p>Added Section 2.1.58 HCI_LE_SET_ADVERTISING_SET_RANDOM_ADDRESS, Section 2.1.59 HCI_LE_SET_EXTENDED_ADVERTISING_PARAMETERS, Section 2.1.60 HCI_LE_SET_EXTENDED_ADVERTISING_DATA, Section 2.1.61 HCI_LE_SET_EXTENDED_SCAN_RESPONSE_DATA, Section 2.1.62 HCI_LE_SET_EXTENDED_ADVERTISING_ENABLE, Section 2.1.63 HCI_LE_READ_MAXIMUM_ADVERTISING_DATA_LENGTH, Section 2.1.64 HCI_LE_READ_NUMBER_OF_SUPPORTED_ADVERTISING_SETS, Section 2.1.65 HCI_LE_REMOVE_ADVERTISING_SET, Section 2.1.66 HCI_LE_CLEAR_ADVERTISING_SETS, Section 2.1.67 HCI_LE_SET_EXTENDED_SCAN_PARAMETERS, Section 2.1.68 HCI_LE_SET_EXTENDED_SCAN_ENABLE, Section 2.1.69 HCI_LE_EXTENDED_CREATE_CONNECTION, Section 2.3.13 ACI_HAL_SET_PERIPHERAL_LATENCY, Section 2.4.42 ACI_GAP_ADD_DEVICES_TO_LIST, Section 2.4.46 ACI_GAP_ADV_SET_CONFIGURATION, Section 2.4.47 ACI_GAP_ADV_SET_ENABLE, Section 2.4.48 ACI_GAP_ADV_SET_ADV_DATA, Section 2.4.49 ACI_GAP_ADV_SET_SCAN_RESP_DATA, Section 2.4.50 ACI_GAP_ADV_REMOVE_SET, Section 2.4.51 ACI_GAP_ADV_CLEAR_SETS, Section 2.4.52 ACI_GAP_ADV_SET_RANDOM_ADDRESS, Section 3.2.12 HCI_LE_EXTENDED_ADVERTISING_REPORT_EVENT, Section 3.2.13 HCI_LE_SCAN_TIMEOUT_EVENT, Section 3.2.14 HCI_LE_ADVERTISING_SET_TERMINATED_EVENT, Section 3.2.15 HCI_LE_SCAN_REQUEST_RECEIVED_EVENT, and Section 3.2.16 HCI_LE_CHANNEL_SELECTION_ALGORITHM_EVENT.</p> <p>Updated Section 2.1.19 HCI_LE_READ_ADVERTISING_PHYSICAL_CHANNEL_TX_POWER, Section 2.1.37 HCI_LE_ENABLE_ENCRYPTION, Section 2.1.39 HCI_LE_LONG_TERM_KEY_REQUEST_NEGATIVE_REPLY, Section 2.2.4 HCI_LE_RECEIVER_TEST_V2, Section 2.2.5 HCI_LE_TRANSMITTER_TEST_V2, Section 2.3.2 ACI_HAL_WRITE_CONFIG_DATA, and Section 2.4.4 ACI_GAP_SET_DIRECT_CONNECTABLE.</p> <p>Removed former Section 2.4.24 ACI_GAP_START_NAME_DISCOVERY_PROC.</p> <p>Minor text edits across the whole document.</p>
29-Apr-2022	12	<p>Updated Table 1. HCI commands list, Table 22. HCI_LE_SET_EVENT_MASK input parameters, Table 129. HCI_LE_TRANSMITTER_TEST input parameters, Table 136. HAL commands list, Table 167. GAP commands list, and Table 393. HCI LE META events command list.</p> <p>Added Section 2.1.71 HCI_LE_READ_RF_PATH_COMPENSATION and Section 2.1.72 HCI_LE_WRITE_RF_PATH_COMPENSATION.</p>
17-Jun-2022	13	<p>Updated Introduction, Section 2.1 HCI commands, Section 2.2 HCI testing commands, Section 2.3 HAL commands, Section 2.4 GAP commands, Section 2.4.10 ACI_GAP_INIT, Section 2.5 GATT/ATT commands, Section 2.6 L2CAP commands, Section 3.1 HCI events, Section 3.2 HCI LE META events, Section 3.3 ACI GAP events, Section 3.4 ACI GATT/ATT events, Section 3.5 ACI L2CAP events, and Section 3.6 ACI HAL events.</p> <p>Updated Table 1. HCI commands list, Table 116. HCI_LE_EXTENDED_CREATE_CONNECTION input parameters, Table 126. HCI testing commands list, Table 136. HAL commands list, Table 167. GAP commands list, Table 270. GATT/ATT commands list, Table 365. L2CAP commands list, Table 384. HCI events commands list, Table 393. HCI LE META events command list, Table 409. ACI GAP events commands list, Table 417. ACI GATT/ATT events list, Table 447. ACI L2CAP events commands list, and Table 459. ACI HAL events commands list.</p> <p>Minor text edits across the whole document.</p>
06-Jul-2022	14	<p>Updated Section 2.5.6 ACI_GATT_UPDATE_CHAR_VALUE, and Section 3.2.10 HCI_LE_DIRECTED_ADVERTISING_REPORT_EVENT.</p> <p>Added Section 2.5.46 ACI_GATT_STORE_DB.</p> <p>Minor text edits across the whole document.</p>

Date	Version	Changes
06-Jul-2022	14 (cont'd)	Updated <a href="#">Table 116. HCI_LE_EXTENDED_CREATE_CONNECTION</a> input parameters, <a href="#">Table 136. HAL commands list</a> , <a href="#">Table 138. ACI_HAL_WRITE_CONFIG_DATA</a> input parameters, <a href="#">Table 148. ACI_HAL_GET_LINK_STATUS</a> output parameters, <a href="#">Table 248. ACI_GAP_ADDITIONAL_BEACON_START</a> input parameters, <a href="#">Table 259. ACI_GAP_ADV_SET_SCAN RESP DATA</a> input parameters, <a href="#">Table 260. ACI_GAP_ADV_SET_SCAN RESP DATA</a> output parameters, <a href="#">Table 263. ACI_GAP_ADV_CLEAR_SETS</a> output parameters, <a href="#">Table 270. GATT/ATT commands list</a> , <a href="#">Table 403. HCI LE DIRECTED_ADVERTISING_REPORT_EVENT</a> parameters, and <a href="#">Table 460. ACI_HAL_END_OF_RADIO_ACTIVITY_EVENT</a> parameters.
21-Jul-2022	15	Updated <a href="#">Table 162. ACI_HAL_READ_RAW_RSSI</a> output parameters.
16-Aug-2022	16	Updated <a href="#">Table 6. HCI_SET_EVENT_MASK</a> input parameters, <a href="#">Table 22. HCI LE SET EVENT MASK</a> input parameters, <a href="#">Table 97. HCI LE SET_ADVERTISING_SET_RANDOM_ADDRESS</a> input parameters, <a href="#">Table 112. HCI LE SET_EXTENDED_SCAN_PARAMETERS</a> input parameters, <a href="#">Table 116. HCI LE_EXTENDED_CREATE_CONNECTION</a> input parameters, <a href="#">Table 119. HCI LE_READ_RF_PATH_COMPENSATION</a> output parameters, <a href="#">Table 120. HCI LE_WRITE_RF_PATH_COMPENSATION</a> input parameters, <a href="#">Table 136. HAL commands list</a> , <a href="#">Table 209. ACI_GAP_START_GENERAL_DISCOVERY_PROC</a> input parameters, <a href="#">Table 211. ACI_GAP_START_AUTO_CONNECTION_ESTABLISH_PROC</a> input parameters, <a href="#">Table 213. ACI_GAP_START_GENERAL_CONNECTION_ESTABLISH_PROC</a> input parameters, <a href="#">Table 215. ACI_GAP_START_SELECTIVE_CONNECTION_ESTABLISH_PROC</a> input parameters, <a href="#">Table 217. ACI_GAP_CREATE_CONNECTION</a> input parameters, and <a href="#">Table 229. ACI_GAP_START_OBSERVATION_PROC</a> input parameters. Added <a href="#">Section 2.3.14 ACI_HAL_READ_RSSI</a> . Minor text edits across the whole document.
04-Jan-2023	17	Updated <a href="#">Section 2.1.4 HCI_RESET</a> , <a href="#">Section 2.3.5 ACI_HAL_LE_TX_TEST_PACKET_NUMBER</a> , <a href="#">Section 2.3.13 ACI_HAL_SET_PERIPHERAL_LATENCY</a> , <a href="#">Section 2.3.14 ACI_GAP_SET_NON_DISCOVERABLE</a> , <a href="#">Section 2.4.1 ACI_GAP_SET_DISCOVERABLE</a> , <a href="#">Section 2.4.3 ACI_GAP_SET_DISCOVERABLE</a> , <a href="#">Section 2.4.4 ACI_GAP_SET_DIRECT_CONNECTABLE</a> , <a href="#">Section 2.4.12 ACI_GAP_SET_UNDIRECTED_CONNECTABLE</a> , <a href="#">Section 2.4.32 ACI_GAP_SET_BROADCAST_MODE</a> , <a href="#">Section 2.5.2 ACI_GATT_ADD_SERVICE</a> , and <a href="#">Section 2.5.4 ACI_GATT_ADD_CHAR</a> . Added <a href="#">Section 2.5.47 ACI_GATT_SET_MULT_NOTIFICATION</a> , <a href="#">Section 2.5.48 ACI_GATT_READ_MULTIPLE_VAR_CHAR_VALUE</a> , <a href="#">Section 3.4.24 ACI_GATT_EATT_BEARER_EVENT</a> , <a href="#">Section 3.4.25 ACI_GATT_MULT_NOTIFICATION_EVENT</a> , and <a href="#">Section 5 Tx power level</a> . Minor text edits across the whole document.

Date	Version	Changes
04-Jan-2023	17 (cont'd)	Updated Table 13. HCI_HOST_BUFFER_SIZE input parameters, Table 119. HCI_LE_READ_RF_PATH_COMPENSATION output parameters, Table 136. HAL commands list, Table 138. ACI_HAL_WRITE_CONFIG_DATA input parameters, Table 145. ACI_HAL_TONE_START input parameters, Table 148. ACI_HAL_GET_LINK_STATUS output parameters, Table 163. ACI_HAL_RX_START input parameters, Table 270. GATT/ATT commands list, Table 272. ACI_GATT_ADD_SERVICE input parameters, Table 276. ACI_GATT_ADD_CHAR input parameters, Table 304. ACI_GATT_DISC_ALL_PRIMARY_SERVICES input parameters, Table 306. ACI_GATT_DISC_PRIMARY_SERVICE_BY_UUID input parameters, Table 308. ACI_GATT_FIND_INCLUDED_SERVICES input parameters, Table 312. ACI_GATT_DISC_CHAR_BY_UUID input parameters, Table 314. ACI_GATT_DISC_ALL_CHAR_DESC input parameters, Table 316. ACI_GATT_READ_CHAR_VALUE input parameters, Table 318. ACI_GATT_READ_USING_CHAR_UUID input parameters, Table 320. ACI_GATT_READ_LONG_CHAR_VALUE input parameters, Table 322. ACI_GATT_READ_MULTIPLE_CHAR_VALUE input parameters, Table 324. ACI_GATT_WRITE_CHAR_VALUE input parameters, Table 326. ACI_GATT_WRITE_LONG_CHAR_VALUE input parameters, Table 328. ACI_GATT_WRITE_CHAR_RELIABLE input parameters, Table 330. ACI_GATT_WRITE_LONG_CHAR_DESC input parameters, Table 332. ACI_GATT_READ_LONG_CHAR_DESC input parameters, Table 334. ACI_GATT_WRITE_CHAR_DESC input parameters, Table 336. ACI_GATT_READ_CHAR_DESC input parameters, Table 338. ACI_GATT_WRITE_WITHOUT_RESP input parameters, Table 340. ACI_GATT_SIGNED_WRITE_WITHOUT_RESP input parameters, Table 342. ACI_GATT_CONFIRM_INDICATION input parameters, Table 344. ACI_GATT_WRITE_RESP input parameters, Table 346. ACI_GATT_ALLOW_READ input parameters, Table 356. ACI_GATT_DENY_READ input parameters, Table 417. ACI GATT/ATT events list, Table 418. ACI_GATT_ATTRIBUTE_MODIFIED_EVENT parameters, Table 419. ACI_GATT_PROC_TIMEOUT_EVENT parameters, Table 421 to Table 429, Table 432 to Table 438, Table 440. ACI_GATT_PREPARE_WRITE_PERMIT_REQ_EVENT parameters, Table 446. ACI_GATT_NOTIFICATION_EXT_EVENT parameters, and Table 448. ACI_L2CAP_CONNECTION_UPDATE_RESP_EVENT parameters.
19-Jan-2023	18	Updated Table 354. ACI_GATT_UPDATE_CHAR_VALUE_EXT input parameters, Table 388. HCI_HARDWARE_ERROR_EVENT parameters, Table 430. ACI_GATT_INDICATION_EVENT parameters, Table 431. ACI_GATT_NOTIFICATION_EVENT parameters, Table 433. ACI_GATT_ERROR_RESP_EVENT parameters, Table 439. ACI_GATT_SERVER_CONFIRMATION_EVENT parameters, Table 444. ACI_GATT_READ_EXT_EVENT parameters, and Table 445. ACI_GATT_INDICATION_EXT_EVENT parameters. Updated Section 3.3.8 ACI_GAP_ADDR_NOT_RESOLVED_EVENT.
23-Feb-2023	19	Updated Table 1. HCI commands list, Table 393. HCI LE META events command list, and Table 463. Status error codes description. Updated Section 2.1.22 HCI_LE_SET_ADVERTISING_ENABLE and Section 3.1.1 HCI_DISCONNECTION_COMPLETE_EVENT. Minor text edits across the whole document.
12-May-2023	20	Updated Section Introduction, Section 2.1.1 HCI_DISCONNECT to Section 2.1.73 HCI_LE_SET_PRIVACY_MODE, Section 2.2.1 HCI_LE_RECEIVER_TEST to Section 2.2.5 HCI_LE_TRANSMITTER_TEST_V2, Section 2.3.11 ACI_HAL_SET_EVENT_MASK, Section 2.3.13 ACI_HAL_SET_PERIPHERAL_LATENCY, Section 2.4.4 ACI_GAP_SET_DIRECT_CONNECTABLE, Section 2.4.13 ACI_GAP_PERIPHERAL_SECURITY_REQ, Section 2.4.17 ACI_GAP_SET_EVENT_MASK, Section 2.4.18 ACI_GAP_CONFIGURE_FILTER_ACCEPT_LIST, Section 2.5.4 ACI_GATT_ADD_CHAR, Section 2.5.5 ACI_GATT_ADD_CHAR_DESC, Section 2.5.6 ACI_GATT_UPDATE_CHAR_VALUE, Section 2.5.10 ACI_GATT_SET_EVENT_MASK, Section 2.5.43 ACI_GATT_UPDATE_CHAR_VALUE_EXT, Section 3.1.1 HCI_DISCONNECTION_COMPLETE_EVENT, Section 3.3.5 ACI_GAP_PERIPHERAL_SECURITY_INITIATED_EVENT, and Section 3.1.6 HCI_ENCRYPTION_KEY_REFRESH_COMPLETE_EVENT. Added Section 2.1.74 HCI_LE_GENERATE_DHKEY_V2 and Section 3.4.26 ACI_GATT_NOTIFICATION_COMPLETE_EVENT.

Date	Version	Changes
12-May-2023	20 (cont'd)	<p>Changed "0xEA00 ... 0xEA1F" with "0xEA00 ... 0xEA3F" as range of possible values for Connection_Handle parameter throughout the whole document.</p> <p>Updated Table 1. HCI commands list, Table 136. HAL commands list, Table 138. ACI_HAL_WRITE_CONFIG_DATA input parameters, Table 167. GAP commands list, Table 199. ACI_GAP_SET_EVENT_MASK input parameters, Table 238. ACI_GAP_GET_OOB_DATA input parameters, Table 239. ACI_GAP_GET_OOB_DATA output parameters, Table 240. ACI_GAP_SET_OOB_DATA input parameters, Table 409. ACI GAP events commands list, and Table 417. ACI GATT/ATT events list.</p> <p>Minor text edits across the whole document.</p>
14-Jul-2023	21	<p>Updated Table 41. HCI_LE_CREATE_CONNECTION input parameters, Table 50. HCI_LE_CONNECTION_UPDATE input parameters, Table 95. HCI_LE_SET_PHY input parameters, Table 116. HCI_LE_EXTENDED_CREATE_CONNECTION input parameters, Table 142. ACI_HAL_SET_TX_POWER_LEVEL input parameters, Table 366. ACI_L2CAP_CONNECTION_PARAMETER_UPDATE_REQ input parameters, Table 368. ACI_L2CAP_CONNECTION_PARAMETER_UPDATE_RESP input parameters, Table 370. ACI_L2CAP_CO_C_CONNECT input parameters, Table 372. ACI_L2CAP_CO_C_CONNECT_CONFIRM input parameters, Table 374. ACI_L2CAP_CO_C_RECONF input parameters, Table 452. ACI_L2CAP_CO_C_CONNECT_EVENT parameters, Table 453. ACI_L2CAP_CO_C_CONNECT_CONFIRM_EVENT parameters, and Table 454. ACI_L2CAP_CO_C_RECONF_EVENT parameters.</p> <p>Updated list of generated events in Section 2.1.25 HCI_LE_CREATE_CONNECTION, Section 2.1.26 HCI_LE_CREATE_CONNECTION_CANCEL, Section 2.1.62 HCI_LE_SET_EXTENDED_ADVERTISING_ENABLE, Section 2.1.69 HCI_LE_EXTENDED_CREATE_CONNECTION, Section 2.4.26 ACI_GAP_START_SELECTIVE_CONNECTION_ESTABLISH_PROC, and Section 2.4.27 ACI_GAP_CREATE_CONNECTION.</p> <p>Updated Section 2.3.4 ACI_HAL_SET_TX_POWER_LEVEL, Section 2.4.20 ACI_GAP_CLEAR_SECURITY_DB, Section 2.4.27 ACI_GAP_CREATE_CONNECTION, and Section 2.5.6 ACI_GATT_UPDATE_CHAR_VALUE.</p> <p>Minor text edits across the whole document.</p>
03-Oct-2023	22	<p>Updated Section 2.4.46 ACI_GAP_ADV_SET_CONFIGURATION.</p> <p>Updated Table 99. HCI_LE_SET_EXTENDED_ADVERTISING_PARAMETERS input parameters, Table 112. HCI_LE_SET_EXTENDED_SCAN_PARAMETERS input parameters, Table 132. HCI_LE_RECEIVER_TEST_V2 input parameters, and Table 253. ACI_GAP_ADV_SET_CONFIGURATION input parameters.</p>
10-Nov-2023	23	<p>Updated Table 16. HCI_READ_LOCAL_VERSION_INFORMATION output parameters, Table 93. HCI_LE_SET_DEFAULT_PHY input parameters, Table 95. HCI_LE_SET_PHY input parameters, Table 167. GAP commands list, Table 276. ACI_GATT_ADD_CHAR input parameters, Table 278. ACI_GATT_ADD_CHAR_DESC input parameters, Table 462. ACI_HAL_FW_ERROR_EVENT parameters, and Table 463. Status error codes description.</p> <p>Added Section 2.4.53 ACI_GAP_EXT_START_SCAN and Section 2.4.54 ACI_GAP_EXT_CREATE_CONNECTION.</p> <p>Updated Section 2.4.35 ACI_GAP_IS_DEVICE_BONDED and Section 2.4.41 ACI_GAP_REMOVE_BONDED_DEVICE.</p>

## Contents

<b>1</b>	<b>General information</b>	<b>2</b>
<b>2</b>	<b>ACI/HCI commands</b>	<b>3</b>
<b>2.1</b>	<b>HCI commands</b>	<b>3</b>
<b>2.1.1</b>	<b>HCI_DISCONNECT</b>	<b>5</b>
<b>2.1.2</b>	<b>HCI_READ_REMOTE_VERSION_INFORMATION</b>	<b>5</b>
<b>2.1.3</b>	<b>HCI_SET_EVENT_MASK</b>	<b>5</b>
<b>2.1.4</b>	<b>HCI_RESET</b>	<b>6</b>
<b>2.1.5</b>	<b>HCI_READ_TRANSMIT_POWER_LEVEL</b>	<b>6</b>
<b>2.1.6</b>	<b>HCI_SET_CONTROLLER_TO_HOST_FLOW_CONTROL</b>	<b>7</b>
<b>2.1.7</b>	<b>HCI_HOST_BUFFER_SIZE</b>	<b>7</b>
<b>2.1.8</b>	<b>HCI_HOST_NUMBER_OF_COMPLETED_PACKETS</b>	<b>8</b>
<b>2.1.9</b>	<b>HCI_READ_LOCAL_VERSION_INFORMATION</b>	<b>9</b>
<b>2.1.10</b>	<b>HCI_READ_LOCAL_SUPPORTED_COMMANDS</b>	<b>9</b>
<b>2.1.11</b>	<b>HCI_READ_LOCAL_SUPPORTED_FEATURES</b>	<b>10</b>
<b>2.1.12</b>	<b>HCI_READ_BD_ADDR</b>	<b>10</b>
<b>2.1.13</b>	<b>HCI_READ_RSSI</b>	<b>10</b>
<b>2.1.14</b>	<b>HCI_LE_SET_EVENT_MASK</b>	<b>11</b>
<b>2.1.15</b>	<b>HCI_LE_READ_BUFFER_SIZE</b>	<b>12</b>
<b>2.1.16</b>	<b>HCI_LE_READ_LOCAL_SUPPORTED_FEATURES</b>	<b>12</b>
<b>2.1.17</b>	<b>HCI_LE_SET_RANDOM_ADDRESS</b>	<b>13</b>
<b>2.1.18</b>	<b>HCI_LE_SET_ADVERTISING_PARAMETERS</b>	<b>13</b>
<b>2.1.19</b>	<b>HCI_LE_READ_ADVERTISING_PHYSICAL_CHANNEL_TX_POWER</b>	<b>14</b>
<b>2.1.20</b>	<b>HCI_LE_SET_ADVERTISING_DATA</b>	<b>15</b>
<b>2.1.21</b>	<b>HCI_LE_SET_SCAN_RESPONSE_DATA</b>	<b>15</b>
<b>2.1.22</b>	<b>HCI_LE_SET_ADVERTISING_ENABLE</b>	<b>15</b>
<b>2.1.23</b>	<b>HCI_LE_SET_SCAN_PARAMETERS</b>	<b>16</b>
<b>2.1.24</b>	<b>HCI_LE_SET_SCAN_ENABLE</b>	<b>17</b>
<b>2.1.25</b>	<b>HCI_LE_CREATE_CONNECTION</b>	<b>17</b>
<b>2.1.26</b>	<b>HCI_LE_CREATE_CONNECTION_CANCEL</b>	<b>19</b>
<b>2.1.27</b>	<b>HCI_LE_READ_FILTER_ACCEPT_LIST_SIZE</b>	<b>19</b>
<b>2.1.28</b>	<b>HCI_LE_CLEAR_FILTER_ACCEPT_LIST</b>	<b>19</b>
<b>2.1.29</b>	<b>HCI_LE_ADD_DEVICE_TO_FILTER_ACCEPT_LIST</b>	<b>20</b>
<b>2.1.30</b>	<b>HCI_LE_REMOVE_DEVICE_FROM_FILTER_ACCEPT_LIST</b>	<b>20</b>
<b>2.1.31</b>	<b>HCI_LE_CONNECTION_UPDATE</b>	<b>21</b>
<b>2.1.32</b>	<b>HCI_LE_SET_HOST_CHANNEL_CLASSIFICATION</b>	<b>22</b>
<b>2.1.33</b>	<b>HCI_LE_READ_CHANNEL_MAP</b>	<b>22</b>

<b>2.1.34</b>	HCI_LE_READ_REMOTE_FEATURES .....	22
<b>2.1.35</b>	HCI_LE_ENCRYPT .....	23
<b>2.1.36</b>	HCI_LE_RAND .....	23
<b>2.1.37</b>	HCI_LE_ENABLE_ENCRYPTION .....	24
<b>2.1.38</b>	HCI_LE_LONG_TERM_KEY_REQUEST_REPLY .....	24
<b>2.1.39</b>	HCI_LE_LONG_TERM_KEY_REQUEST_NEGATIVE_REPLY .....	24
<b>2.1.40</b>	HCI_LE_READ_SUPPORTED_STATES .....	25
<b>2.1.41</b>	HCI_LE_SET_DATA_LENGTH .....	25
<b>2.1.42</b>	HCI_LE_READ_SUGGESTED_DEFAULT_DATA_LENGTH .....	26
<b>2.1.43</b>	HCI_LE_WRITE_SUGGESTED_DEFAULT_DATA_LENGTH .....	26
<b>2.1.44</b>	HCI_LE_READ_LOCAL_P256_PUBLIC_KEY .....	26
<b>2.1.45</b>	HCI_LE_GENERATE_DHKEY .....	27
<b>2.1.46</b>	HCI_LE_ADD_DEVICE_TO_RESOLVING_LIST .....	27
<b>2.1.47</b>	HCI_LE_REMOVE_DEVICE_FROM_RESOLVING_LIST .....	28
<b>2.1.48</b>	HCI_LE_CLEAR_RESOLVING_LIST .....	28
<b>2.1.49</b>	HCI_LE_READ_RESOLVING_LIST_SIZE .....	28
<b>2.1.50</b>	HCI_LE_READ_PEER_RESOLVABLE_ADDRESS .....	29
<b>2.1.51</b>	HCI_LE_READ_LOCAL_RESOLVABLE_ADDRESS .....	29
<b>2.1.52</b>	HCI_LE_SET_ADDRESS_RESOLUTION_ENABLE .....	30
<b>2.1.53</b>	HCI_LE_SET_RESOLVABLE_PRIVATE_ADDRESS_TIMEOUT .....	30
<b>2.1.54</b>	HCI_LE_READ_MAXIMUM_DATA_LENGTH .....	31
<b>2.1.55</b>	HCI_LE_READ_PHY .....	31
<b>2.1.56</b>	HCI_LE_SET_DEFAULT_PHY .....	31
<b>2.1.57</b>	HCI_LE_SET_PHY .....	32
<b>2.1.58</b>	HCI_LE_SET_ADVERTISING_SET_RANDOM_ADDRESS .....	33
<b>2.1.59</b>	HCI_LE_SET_EXTENDED_ADVERTISING_PARAMETERS .....	34
<b>2.1.60</b>	HCI_LE_SET_EXTENDED_ADVERTISING_DATA .....	35
<b>2.1.61</b>	HCI_LE_SET_EXTENDED_SCAN_RESPONSE_DATA .....	36
<b>2.1.62</b>	HCI_LE_SET_EXTENDED_ADVERTISING_ENABLE .....	37
<b>2.1.63</b>	HCI_LE_READ_MAXIMUM_ADVERTISING_DATA_LENGTH .....	37
<b>2.1.64</b>	HCI_LE_READ_NUMBER_OF_SUPPORTED_ADVERTISING_SETS .....	37
<b>2.1.65</b>	HCI_LE_REMOVE_ADVERTISING_SET .....	38
<b>2.1.66</b>	HCI_LE_CLEAR_ADVERTISING_SETS .....	38
<b>2.1.67</b>	HCI_LE_SET_EXTENDED_SCAN_PARAMETERS .....	38
<b>2.1.68</b>	HCI_LE_SET_EXTENDED_SCAN_ENABLE .....	39
<b>2.1.69</b>	HCI_LE_EXTENDED_CREATE_CONNECTION .....	40
<b>2.1.70</b>	HCI_LE_READ_TRANSMIT_POWER .....	41
<b>2.1.71</b>	HCI_LE_READ_RF_PATH_COMPENSATION .....	41

<b>2.1.72</b>	HCI_LE_WRITE_RF_PATH_COMPENSATION .....	42
<b>2.1.73</b>	HCI_LE_SET_PRIVACY_MODE.....	42
<b>2.1.74</b>	HCI_LE_GENERATE_DHKEY_V2 .....	43
<b>2.2</b>	HCI testing commands .....	44
<b>2.2.1</b>	HCI_LE_RECEIVER_TEST .....	44
<b>2.2.2</b>	HCI_LE_TRANSMITTER_TEST .....	44
<b>2.2.3</b>	HCI_TEST_END.....	45
<b>2.2.4</b>	HCI_LE_RECEIVER_TEST_V2 .....	45
<b>2.2.5</b>	HCI_LE_TRANSMITTER_TEST_V2.....	46
<b>2.3</b>	HAL commands.....	47
<b>2.3.1</b>	ACI_HAL_GET_FW_BUILD_NUMBER.....	47
<b>2.3.2</b>	ACI_HAL_WRITE_CONFIG_DATA.....	47
<b>2.3.3</b>	ACI_HAL_READ_CONFIG_DATA.....	48
<b>2.3.4</b>	ACI_HAL_SET_TX_POWER_LEVEL.....	49
<b>2.3.5</b>	ACI_HAL_LE_TX_TEST_PACKET_NUMBER .....	49
<b>2.3.6</b>	ACI_HAL_TONE_START .....	50
<b>2.3.7</b>	ACI_HAL_TONE_STOP .....	50
<b>2.3.8</b>	ACI_HAL_GET_LINK_STATUS .....	50
<b>2.3.9</b>	ACI_HAL_SET_RADIO_ACTIVITY_MASK.....	51
<b>2.3.10</b>	ACI_HAL_GET_ANCHOR_PERIOD.....	51
<b>2.3.11</b>	ACI_HAL_SET_EVENT_MASK .....	52
<b>2.3.12</b>	ACI_HAL_GET_PM_DEBUG_INFO .....	52
<b>2.3.13</b>	ACI_HAL_SET_PERIPHERAL_LATENCY .....	52
<b>2.3.14</b>	ACI_HAL_READ_RSSI.....	53
<b>2.3.15</b>	ACI_HAL_READ_RADIO_REG .....	53
<b>2.3.16</b>	ACI_HAL_WRITE_RADIO_REG .....	53
<b>2.3.17</b>	ACI_HAL_READ_RAW_RSSI.....	54
<b>2.3.18</b>	ACI_HAL_RX_START .....	54
<b>2.3.19</b>	ACI_HAL_RX_STOP .....	54
<b>2.3.20</b>	ACI_HAL_STACK_RESET .....	55
<b>2.4</b>	GAP commands .....	56
<b>2.4.1</b>	ACI_GAP_SET_NON_DISCOVERABLE .....	57
<b>2.4.2</b>	ACI_GAP_SET_LIMITED_DISCOVERABLE .....	57
<b>2.4.3</b>	ACI_GAP_SET_DISCOVERABLE .....	59
<b>2.4.4</b>	ACI_GAP_SET_DIRECT_CONNECTABLE .....	60
<b>2.4.5</b>	ACI_GAP_SET_IO_CAPABILITY .....	61
<b>2.4.6</b>	ACI_GAP_SET_AUTHENTICATION_REQUIREMENT.....	61
<b>2.4.7</b>	ACI_GAP_SET_AUTHORIZATION_REQUIREMENT.....	62

<b>2.4.8</b>	ACI_GAP_PASS_KEY_RESP.....	63
<b>2.4.9</b>	ACI_GAP_AUTHORIZATION_RESP .....	63
<b>2.4.10</b>	ACI_GAP_INIT .....	64
<b>2.4.11</b>	ACI_GAP_SET_NON_CONNECTABLE .....	64
<b>2.4.12</b>	ACI_GAP_SET_UNDIRECTED_CONNECTABLE .....	65
<b>2.4.13</b>	ACI_GAP_PERIPHERAL_SECURITY_REQ .....	65
<b>2.4.14</b>	ACI_GAP_UPDATE_ADV_DATA .....	66
<b>2.4.15</b>	ACI_GAP_DELETE_AD_TYPE .....	66
<b>2.4.16</b>	ACI_GAP_GET_SECURITY_LEVEL .....	66
<b>2.4.17</b>	ACI_GAP_SET_EVENT_MASK .....	67
<b>2.4.18</b>	ACI_GAP_CONFIGURE_FILTER_ACCEPT_LIST .....	67
<b>2.4.19</b>	ACI_GAP_TERMINATE .....	68
<b>2.4.20</b>	ACI_GAP_CLEAR_SECURITY_DB .....	68
<b>2.4.21</b>	ACI_GAP_ALLOW_REBOND.....	68
<b>2.4.22</b>	ACI_GAP_START_LIMITED_DISCOVERY_PROC .....	69
<b>2.4.23</b>	ACI_GAP_START_GENERAL_DISCOVERY_PROC .....	70
<b>2.4.24</b>	ACI_GAP_START_AUTO_CONNECTION_ESTABLISH_PROC .....	70
<b>2.4.25</b>	ACI_GAP_START_GENERAL_CONNECTION_ESTABLISH_PROC .....	72
<b>2.4.26</b>	ACI_GAP_START_SELECTIVE_CONNECTION_ESTABLISH_PROC .....	73
<b>2.4.27</b>	ACI_GAP_CREATE_CONNECTION.....	74
<b>2.4.28</b>	ACI_GAP_TERMINATE_GAP_PROC.....	75
<b>2.4.29</b>	ACI_GAP_START_CONNECTION_UPDATE .....	76
<b>2.4.30</b>	ACI_GAP_SEND_PAIRING_REQ.....	77
<b>2.4.31</b>	ACI_GAP_RESOLVE_PRIVATE_ADDR.....	77
<b>2.4.32</b>	ACI_GAP_SET_BROADCAST_MODE .....	77
<b>2.4.33</b>	ACI_GAP_START_OBSERVATION_PROC .....	78
<b>2.4.34</b>	ACI_GAP_GET_BONDED_DEVICES .....	80
<b>2.4.35</b>	ACI_GAP_IS_DEVICE_BONDED.....	80
<b>2.4.36</b>	ACI_GAP_NUMERIC_COMPARISON_VALUE_CONFIRM_YESNO .....	80
<b>2.4.37</b>	ACI_GAP_PASSKEY_INPUT .....	81
<b>2.4.38</b>	ACI_GAP_GET_OOB_DATA .....	81
<b>2.4.39</b>	ACI_GAP_SET_OOB_DATA.....	82
<b>2.4.40</b>	ACI_GAP_ADD_DEVICES_TO_RESOLVING_LIST.....	82
<b>2.4.41</b>	ACI_GAP_REMOVE_BONDED_DEVICE.....	83
<b>2.4.42</b>	ACI_GAP_ADD_DEVICES_TO_LIST.....	83
<b>2.4.43</b>	ACI_GAP_ADDITIONAL_BEACON_START.....	84
<b>2.4.44</b>	ACI_GAP_ADDITIONAL_BEACON_STOP.....	85
<b>2.4.45</b>	ACI_GAP_ADDITIONAL_BEACON_SET_DATA.....	85

<b>2.4.46</b>	ACI_GAP_ADV_SET_CONFIGURATION .....	85
<b>2.4.47</b>	ACI_GAP_ADV_SET_ENABLE .....	87
<b>2.4.48</b>	ACI_GAP_ADV_SET_ADV_DATA .....	87
<b>2.4.49</b>	ACI_GAP_ADV_SET_SCAN_RESP_DATA .....	88
<b>2.4.50</b>	ACI_GAP_ADV_REMOVE_SET .....	89
<b>2.4.51</b>	ACI_GAP_ADV_CLEAR_SETS .....	89
<b>2.4.52</b>	ACI_GAP_ADV_SET_RANDOM_ADDRESS .....	89
<b>2.4.53</b>	ACI_GAP_EXT_START_SCAN .....	90
<b>2.4.54</b>	ACI_GAP_EXT_CREATE_CONNECTION .....	92
<b>2.5</b>	GATT/ATT commands .....	94
<b>2.5.1</b>	ACI_GATT_INIT .....	95
<b>2.5.2</b>	ACI_GATT_ADD_SERVICE .....	95
<b>2.5.3</b>	ACI_GATT_INCLUDE_SERVICE .....	96
<b>2.5.4</b>	ACI_GATT_ADD_CHAR .....	96
<b>2.5.5</b>	ACI_GATT_ADD_CHAR_DESC .....	98
<b>2.5.6</b>	ACI_GATT_UPDATE_CHAR_VALUE .....	99
<b>2.5.7</b>	ACI_GATT_DEL_CHAR .....	100
<b>2.5.8</b>	ACI_GATT_DEL_SERVICE .....	100
<b>2.5.9</b>	ACI_GATT_DEL_INCLUDE_SERVICE .....	101
<b>2.5.10</b>	ACI_GATT_SET_EVENT_MASK .....	101
<b>2.5.11</b>	ACI_GATT_EXCHANGE_CONFIG .....	102
<b>2.5.12</b>	ACI_ATT_FIND_INFO_REQ .....	102
<b>2.5.13</b>	ACI_ATT_FIND_BY_TYPE_VALUE_REQ .....	103
<b>2.5.14</b>	ACI_ATT_READ_BY_TYPE_REQ .....	103
<b>2.5.15</b>	ACI_ATT_READ_BY_GROUP_TYPE_REQ .....	104
<b>2.5.16</b>	ACI_ATT_PREPARE_WRITE_REQ .....	104
<b>2.5.17</b>	ACI_ATT_EXECUTE_WRITE_REQ .....	105
<b>2.5.18</b>	ACI_GATT_DISC_ALL_PRIMARY_SERVICES .....	105
<b>2.5.19</b>	ACI_GATT_DISC_PRIMARY_SERVICE_BY_UUID .....	106
<b>2.5.20</b>	ACI_GATT_FIND_INCLUDED_SERVICES .....	106
<b>2.5.21</b>	ACI_GATT_DISC_ALL_CHAR_OF_SERVICE .....	107
<b>2.5.22</b>	ACI_GATT_DISC_CHAR_BY_UUID .....	107
<b>2.5.23</b>	ACI_GATT_DISC_ALL_CHAR_DESC .....	108
<b>2.5.24</b>	ACI_GATT_READ_CHAR_VALUE .....	108
<b>2.5.25</b>	ACI_GATT_READ_USING_CHAR_UUID .....	109
<b>2.5.26</b>	ACI_GATT_READ_LONG_CHAR_VALUE .....	110
<b>2.5.27</b>	ACI_GATT_READ_MULTIPLE_CHAR_VALUE .....	110
<b>2.5.28</b>	ACI_GATT_WRITE_CHAR_VALUE .....	111

<b>2.5.29</b>	ACI_GATT_WRITE_LONG_CHAR_VALUE .....	111
<b>2.5.30</b>	ACI_GATT_WRITE_CHAR_RELIABLE .....	112
<b>2.5.31</b>	ACI_GATT_WRITE_LONG_CHAR_DESC .....	113
<b>2.5.32</b>	ACI_GATT_READ_LONG_CHAR_DESC .....	113
<b>2.5.33</b>	ACI_GATT_WRITE_CHAR_DESC .....	114
<b>2.5.34</b>	ACI_GATT_READ_CHAR_DESC .....	114
<b>2.5.35</b>	ACI_GATT_WRITE_WITHOUT_RESP .....	115
<b>2.5.36</b>	ACI_GATT_SIGNED_WRITE_WITHOUT_RESP .....	115
<b>2.5.37</b>	ACI_GATT_CONFIRM_INDICATION .....	116
<b>2.5.38</b>	ACI_GATT_WRITE_RESP .....	116
<b>2.5.39</b>	ACI_GATT_ALLOW_READ .....	117
<b>2.5.40</b>	ACI_GATT_SET_SECURITY_PERMISSION .....	118
<b>2.5.41</b>	ACI_GATT_SET_DESC_VALUE .....	118
<b>2.5.42</b>	ACI_GATT_READ_HANDLE_VALUE .....	119
<b>2.5.43</b>	ACI_GATT_UPDATE_CHAR_VALUE_EXT .....	119
<b>2.5.44</b>	ACI_GATT_DENY_READ .....	120
<b>2.5.45</b>	ACI_GATT_SET_ACCESS_PERMISSION .....	121
<b>2.5.46</b>	ACI_GATT_STORE_DB .....	121
<b>2.5.47</b>	ACI_GATT_SET_MULT_NOTIFICATION .....	121
<b>2.5.48</b>	ACI_GATT_READ_MULTIPLE_VAR_CHAR_VALUE .....	122
<b>2.6</b>	L2CAP commands .....	123
<b>2.6.1</b>	ACI_L2CAP_CONNECTION_PARAMETER_UPDATE_REQ .....	123
<b>2.6.2</b>	ACI_L2CAP_CONNECTION_PARAMETER_UPDATE_RESP .....	123
<b>2.6.3</b>	ACI_L2CAP_CO_C_CONNECT .....	124
<b>2.6.4</b>	ACI_L2CAP_CO_C_CONNECT_CONFIRM .....	125
<b>2.6.5</b>	ACI_L2CAP_CO_C_RECONF .....	125
<b>2.6.6</b>	ACI_L2CAP_CO_C_RECONF_CONFIRM .....	126
<b>2.6.7</b>	ACI_L2CAP_CO_C_DISCONNECT .....	126
<b>2.6.8</b>	ACI_L2CAP_CO_C_FLOW_CONTROL .....	127
<b>2.6.9</b>	ACI_L2CAP_CO_C_TX_DATA .....	127
<b>3</b>	<b>ACI/HCI events .....</b>	<b>128</b>
<b>3.1</b>	HCI events .....	128
<b>3.1.1</b>	HCI_DISCONNECTION_COMPLETE_EVENT .....	128
<b>3.1.2</b>	HCI_ENCRYPTION_CHANGE_EVENT .....	128
<b>3.1.3</b>	HCI_READ_REMOTE_VERSION_INFORMATION_COMPLETE_EVENT .....	129
<b>3.1.4</b>	HCI_HARDWARE_ERROR_EVENT .....	129
<b>3.1.5</b>	HCI_NUMBER_OF_COMPLETED_PACKETS_EVENT .....	130
<b>3.1.6</b>	HCI_ENCRYPTION_KEY_REFRESH_COMPLETE_EVENT .....	130

3.1.7	HCI_COMMAND_COMPLETE_EVENT .....	130
3.1.8	HCI_COMMAND_STATUS_EVENT .....	131
3.2	HCI LE META events .....	132
3.2.1	HCI_LE_CONNECTION_COMPLETE_EVENT .....	132
3.2.2	HCI_LE_ADVERTISING_REPORT_EVENT.....	133
3.2.3	HCI_LE_CONNECTION_UPDATE_COMPLETE_EVENT .....	133
3.2.4	HCI_LE_READ_REMOTE_FEATURES_COMPLETE_EVENT.....	134
3.2.5	HCI_LE_LONG_TERM_KEY_REQUEST_EVENT .....	134
3.2.6	HCI_LE_DATA_LENGTH_CHANGE_EVENT .....	134
3.2.7	HCI_LE_READ_LOCAL_P256_PUBLIC_KEY_COMPLETE_EVENT.....	135
3.2.8	HCI_LE_GENERATE_DHKEY_COMPLETE_EVENT.....	135
3.2.9	HCI_LE_ENHANCED_CONNECTION_COMPLETE_EVENT.....	135
3.2.10	HCI_LE_DIRECTED_ADVERTISING_REPORT_EVENT .....	136
3.2.11	HCI_LE_PHY_UPDATE_COMPLETE_EVENT.....	137
3.2.12	HCI_LE_EXTENDED_ADVERTISING_REPORT_EVENT .....	137
3.2.13	HCI_LE_SCAN_TIMEOUT_EVENT .....	138
3.2.14	HCI_LE_ADVERTISING_SET_TERMINATED_EVENT .....	139
3.2.15	HCI_LE_SCAN_REQUEST_RECEIVED_EVENT.....	139
3.2.16	HCI_LE_CHANNEL_SELECTION_ALGORITHM_EVENT .....	139
3.3	ACI GAP events .....	140
3.3.1	ACI_GAP_LIMITED_DISCOVERABLE_EVENT.....	140
3.3.2	ACI_GAP_PAIRING_COMPLETE_EVENT.....	140
3.3.3	ACI_GAP_PASS_KEY_REQ_EVENT.....	141
3.3.4	ACI_GAP_AUTHORIZATION_REQ_EVENT .....	141
3.3.5	ACI_GAP_PERIPHERAL_SECURITY_INITIATED_EVENT .....	141
3.3.6	ACI_GAP_BOND_LOST_EVENT.....	141
3.3.7	ACI_GAP_PROC_COMPLETE_EVENT.....	141
3.3.8	ACI_GAP_ADDR_NOT_RESOLVED_EVENT .....	142
3.3.9	ACI_GAP_NUMERIC_COMPARISON_VALUE_EVENT.....	142
3.3.10	ACI_GAP_KEYPRESS_NOTIFICATION_EVENT.....	142
3.4	ACI GATT/ATT events .....	143
3.4.1	ACI_GATT_ATTRIBUTE_MODIFIED_EVENT .....	144
3.4.2	ACI_GATT_PROC_TIMEOUT_EVENT.....	144
3.4.3	ACI_ATT_EXCHANGE_MTU_RESP_EVENT.....	144
3.4.4	ACI_ATT_FIND_INFO_RESP_EVENT .....	145
3.4.5	ACI_ATT_FIND_BY_TYPE_VALUE_RESP_EVENT.....	145
3.4.6	ACI_ATT_READ_BY_TYPE_RESP_EVENT .....	145
3.4.7	ACI_ATT_READ_RESP_EVENT .....	146

3.4.8	ACI_ATT_READ_BLOB_RESP_EVENT .....	146
3.4.9	ACI_ATT_READ_MULTIPLE_RESP_EVENT .....	146
3.4.10	ACI_ATT_READ_BY_GROUP_TYPE_RESP_EVENT .....	147
3.4.11	ACI_ATT_PREPARE_WRITE_RESP_EVENT .....	147
3.4.12	ACI_ATT_EXEC_WRITE_RESP_EVENT .....	148
3.4.13	ACI_GATT_INDICATION_EVENT .....	148
3.4.14	ACI_GATT_NOTIFICATION_EVENT .....	148
3.4.15	ACI_GATT_PROC_COMPLETE_EVENT .....	149
3.4.16	ACI_GATT_ERROR_RESP_EVENT .....	149
3.4.17	ACI_GATT_DISC_READ_CHAR_BY_UUID_RESP_EVENT .....	150
3.4.18	ACI_GATT_WRITE_PERMIT_REQ_EVENT .....	151
3.4.19	ACI_GATT_READ_PERMIT_REQ_EVENT .....	151
3.4.20	ACI_GATT_READ_MULTI_PERMIT_REQ_EVENT .....	151
3.4.21	ACI_GATT_TX_POOL_AVAILABLE_EVENT .....	152
3.4.22	ACI_GATT_SERVER_CONFIRMATION_EVENT .....	152
3.4.23	ACI_GATT_PREPARE_WRITE_PERMIT_REQ_EVENT .....	152
3.4.24	ACI_GATT_EATT_BEARER_EVENT .....	153
3.4.25	ACI_GATT_MULT_NOTIFICATION_EVENT .....	153
3.4.26	ACI_GATT_NOTIFICATION_COMPLETE_EVENT .....	154
3.4.27	ACI_GATT_READ_EXT_EVENT .....	154
3.4.28	ACI_GATT_INDICATION_EXT_EVENT .....	154
3.4.29	ACI_GATT_NOTIFICATION_EXT_EVENT .....	155
3.5	ACI L2CAP events .....	156
3.5.1	ACI_L2CAP_CONNECTION_UPDATE_RESP_EVENT .....	156
3.5.2	ACI_L2CAP_PROC_TIMEOUT_EVENT .....	156
3.5.3	ACI_L2CAP_CONNECTION_UPDATE_REQ_EVENT .....	156
3.5.4	ACI_L2CAP_COMMAND_REJECT_EVENT .....	157
3.5.5	ACI_L2CAP_CO_C_CONNECT_EVENT .....	157
3.5.6	ACI_L2CAP_CO_C_CONNECT_CONFIRM_EVENT .....	157
3.5.7	ACI_L2CAP_CO_C_RECONF_EVENT .....	158
3.5.8	ACI_L2CAP_CO_C_RECONF_CONFIRM_EVENT .....	158
3.5.9	ACI_L2CAP_CO_C_DISCONNECT_EVENT .....	158
3.5.10	ACI_L2CAP_CO_C_FLOW_CONTROL_EVENT .....	159
3.5.11	ACI_L2CAP_CO_C_RX_DATA_EVENT .....	159
3.5.12	ACI_L2CAP_CO_C_TX_POOL_AVAILABLE_EVENT .....	159
3.6	ACI HAL events .....	160
3.6.1	ACI_HAL_END_OF_RADIO_ACTIVITY_EVENT .....	160
3.6.2	ACI_HAL_SCAN_REQ_REPORT_EVENT .....	160

---

3.6.3	ACI_HAL_FW_ERROR_EVENT .....	161
<b>4</b>	<b>Status error codes .....</b>	<b>162</b>
<b>5</b>	<b>Tx power level .....</b>	<b>164</b>
<b>Revision history .....</b>		<b>165</b>

## List of tables

<b>Table 1.</b>	HCI commands list . . . . .	3
<b>Table 2.</b>	HCI_DISCONNECT input parameters . . . . .	5
<b>Table 3.</b>	HCI_DISCONNECT output parameters . . . . .	5
<b>Table 4.</b>	HCI_READ_REMOTE_VERSION_INFORMATION input parameters . . . . .	5
<b>Table 5.</b>	HCI_READ_REMOTE_VERSION_INFORMATION output parameters . . . . .	5
<b>Table 6.</b>	HCI_SET_EVENT_MASK input parameters . . . . .	6
<b>Table 7.</b>	HCI_SET_EVENT_MASK output parameters . . . . .	6
<b>Table 8.</b>	HCI_RESET output parameters . . . . .	6
<b>Table 9.</b>	HCI_READ_TRANSMIT_POWER_LEVEL input parameters . . . . .	6
<b>Table 10.</b>	HCI_READ_TRANSMIT_POWER_LEVEL output parameters . . . . .	7
<b>Table 11.</b>	HCI_SET_CONTROLLER_TO_HOST_FLOW_CONTROL input parameters . . . . .	7
<b>Table 12.</b>	HCI_SET_CONTROLLER_TO_HOST_FLOW_CONTROL output parameters . . . . .	7
<b>Table 13.</b>	HCI_HOST_BUFFER_SIZE input parameters . . . . .	8
<b>Table 14.</b>	HCI_HOST_BUFFER_SIZE output parameters . . . . .	8
<b>Table 15.</b>	HCI_HOST_NUMBER_OF_COMPLETED_PACKETS input parameters . . . . .	9
<b>Table 16.</b>	HCI_READ_LOCAL_VERSION_INFORMATION output parameters . . . . .	9
<b>Table 17.</b>	HCI_READ_LOCAL_SUPPORTED_COMMANDS output parameters . . . . .	10
<b>Table 18.</b>	HCI_READ_LOCAL_SUPPORTED_FEATURES output parameters . . . . .	10
<b>Table 19.</b>	HCI_READ_BD_ADDR output parameters . . . . .	10
<b>Table 20.</b>	HCI_READ_RSSI input parameters . . . . .	10
<b>Table 21.</b>	HCI_READ_RSSI output parameters . . . . .	11
<b>Table 22.</b>	HCI_LE_SET_EVENT_MASK input parameters . . . . .	11
<b>Table 23.</b>	HCI_LE_SET_EVENT_MASK output parameters . . . . .	11
<b>Table 24.</b>	HCI_LE_READ_BUFFER_SIZE output parameters . . . . .	12
<b>Table 25.</b>	HCI_LE_READ_LOCAL_SUPPORTED_FEATURES output parameters . . . . .	12
<b>Table 26.</b>	HCI_LE_SET_RANDOM_ADDRESS input parameters . . . . .	13
<b>Table 27.</b>	HCI_LE_SET_RANDOM_ADDRESS output parameters . . . . .	13
<b>Table 28.</b>	HCI_LE_SET_ADVERTISING_PARAMETERS input parameters . . . . .	13
<b>Table 29.</b>	HCI_LE_SET_ADVERTISING_PARAMETERS output parameters . . . . .	14
<b>Table 30.</b>	HCI_LE_READ_ADVERTISING_PHYSICAL_CHANNEL_TX_POWER output parameters . . . . .	14
<b>Table 31.</b>	HCI_LE_SET_ADVERTISING_DATA input parameters . . . . .	15
<b>Table 32.</b>	HCI_LE_SET_ADVERTISING_DATA output parameters . . . . .	15
<b>Table 33.</b>	HCI_LE_SET_SCAN_RESPONSE_DATA input parameters . . . . .	15
<b>Table 34.</b>	HCI_LE_SET_SCAN_RESPONSE_DATA output parameters . . . . .	15
<b>Table 35.</b>	HCI_LE_SET_ADVERTISING_ENABLE input parameters . . . . .	15
<b>Table 36.</b>	HCI_LE_SET_ADVERTISING_ENABLE output parameters . . . . .	16
<b>Table 37.</b>	HCI_LE_SET_SCAN_PARAMETERS input parameters . . . . .	16
<b>Table 38.</b>	HCI_LE_SET_SCAN_PARAMETERS output parameters . . . . .	17
<b>Table 39.</b>	HCI_LE_SET_SCAN_ENABLE input parameters . . . . .	17
<b>Table 40.</b>	HCI_LE_SET_SCAN_ENABLE output parameters . . . . .	17
<b>Table 41.</b>	HCI_LE_CREATE_CONNECTION input parameters . . . . .	18
<b>Table 42.</b>	HCI_LE_CREATE_CONNECTION output parameters . . . . .	19
<b>Table 43.</b>	HCI_LE_CREATE_CONNECTION_CANCEL output parameters . . . . .	19
<b>Table 44.</b>	HCI_LE_READ_FILTER_ACCEPT_LIST_SIZE output parameters . . . . .	19
<b>Table 45.</b>	HCI_LE_CLEAR_FILTER_ACCEPT_LIST output parameters . . . . .	20
<b>Table 46.</b>	HCI_LE_ADD_DEVICE_TO_FILTER_ACCEPT_LIST input parameters . . . . .	20
<b>Table 47.</b>	HCI_LE_ADD_DEVICE_TO_FILTER_ACCEPT_LIST output parameters . . . . .	20
<b>Table 48.</b>	HCI_LE_REMOVE_DEVICE_FROM_FILTER_ACCEPT_LIST input parameters . . . . .	20
<b>Table 49.</b>	HCI_LE_REMOVE_DEVICE_FROM_FILTER_ACCEPT_LIST output parameters . . . . .	20
<b>Table 50.</b>	HCI_LE_CONNECTION_UPDATE input parameters . . . . .	21
<b>Table 51.</b>	HCI_LE_CONNECTION_UPDATE output parameters . . . . .	21
<b>Table 52.</b>	HCI_LE_SET_HOST_CHANNEL_CLASSIFICATION input parameters . . . . .	22
<b>Table 53.</b>	HCI_LE_SET_HOST_CHANNEL_CLASSIFICATION output parameters . . . . .	22

Table 54.	HCI_LE_READ_CHANNEL_MAP input parameters . . . . .	22
Table 55.	HCI_LE_READ_CHANNEL_MAP output parameters . . . . .	22
Table 56.	HCI_LE_READ_REMOTE_FEATURES input parameters . . . . .	23
Table 57.	HCI_LE_READ_REMOTE_FEATURES output parameters . . . . .	23
Table 58.	HCI_LE_ENCRYPT input parameters . . . . .	23
Table 59.	HCI_LE_ENCRYPT output parameters . . . . .	23
Table 60.	HCI_LE_RAND output parameters . . . . .	23
Table 61.	HCI_LE_ENABLE_ENCRYPTION input parameters . . . . .	24
Table 62.	HCI_LE_ENABLE_ENCRYPTION output parameters . . . . .	24
Table 63.	HCI_LE_LONG_TERM_KEY_REQUEST_REPLY input parameters . . . . .	24
Table 64.	HCI_LE_LONG_TERM_KEY_REQUEST_REPLY output parameters . . . . .	24
Table 65.	HCI_LE_LONG_TERM_KEY_REQUEST_NEGATIVE_REPLY input parameters . . . . .	25
Table 66.	HCI_LE_LONG_TERM_KEY_REQUEST_NEGATIVE_REPLY output parameters . . . . .	25
Table 67.	HCI_LE_READ_SUPPORTED_STATES output parameters . . . . .	25
Table 68.	HCI_LE_SET_DATA_LENGTH input parameters . . . . .	25
Table 69.	HCI_LE_SET_DATA_LENGTH output parameters . . . . .	25
Table 70.	HCI_LE_READ_SUGGESTED_DEFAULT_DATA_LENGTH output parameters . . . . .	26
Table 71.	HCI_LE_WRITE_SUGGESTED_DEFAULT_DATA_LENGTH input parameters . . . . .	26
Table 72.	HCI_LE_WRITE_SUGGESTED_DEFAULT_DATA_LENGTH output parameters . . . . .	26
Table 73.	HCI_LE_READ_LOCAL_P256_PUBLIC_KEY output parameters . . . . .	26
Table 74.	HCI_LE_GENERATE_DHKEY input parameters . . . . .	27
Table 75.	HCI_LE_GENERATE_DHKEY output parameters . . . . .	27
Table 76.	HCI_LE_ADD_DEVICE_TO_RESOLVING_LIST input parameters . . . . .	27
Table 77.	HCI_LE_ADD_DEVICE_TO_RESOLVING_LIST output parameters . . . . .	27
Table 78.	HCI_LE_REMOVE_DEVICE_FROM_RESOLVING_LIST input parameters . . . . .	28
Table 79.	HCI_LE_REMOVE_DEVICE_FROM_RESOLVING_LIST output parameters . . . . .	28
Table 80.	HCI_LE_CLEAR_RESOLVING_LIST output parameters . . . . .	28
Table 81.	HCI_LE_READ_RESOLVING_LIST_SIZE output parameters . . . . .	29
Table 82.	HCI_LE_READ_PEER_RESOLVABLE_ADDRESS input parameters . . . . .	29
Table 83.	HCI_LE_READ_PEER_RESOLVABLE_ADDRESS output parameters . . . . .	29
Table 84.	HCI_LE_READ_LOCAL_RESOLVABLE_ADDRESS input parameters . . . . .	29
Table 85.	HCI_LE_READ_LOCAL_RESOLVABLE_ADDRESS output parameters . . . . .	30
Table 86.	HCI_LE_SET_ADDRESS_RESOLUTION_ENABLE input parameters . . . . .	30
Table 87.	HCI_LE_SET_ADDRESS_RESOLUTION_ENABLE output parameters . . . . .	30
Table 88.	HCI_LE_SET_RESOLVABLE_PRIVATE_ADDRESS_TIMEOUT input parameters . . . . .	30
Table 89.	HCI_LE_SET_RESOLVABLE_PRIVATE_ADDRESS_TIMEOUT output parameters . . . . .	30
Table 90.	HCI_LE_READ_MAXIMUM_DATA_LENGTH output parameters . . . . .	31
Table 91.	HCI_LE_READ_PHY input parameters . . . . .	31
Table 92.	HCI_LE_READ_PHY output parameters . . . . .	31
Table 93.	HCI_LE_SET_DEFAULT_PHY input parameters . . . . .	32
Table 94.	HCI_LE_SET_DEFAULT_PHY output parameters . . . . .	32
Table 95.	HCI_LE_SET_PHY input parameters . . . . .	33
Table 96.	HCI_LE_SET_PHY output parameters . . . . .	33
Table 97.	HCI_LE_SET_ADVERTISING_SET_RANDOM_ADDRESS input parameters . . . . .	33
Table 98.	HCI_LE_SET_ADVERTISING_SET_RANDOM_ADDRESS output parameters . . . . .	33
Table 99.	HCI_LE_SET_EXTENDED_ADVERTISING_PARAMETERS input parameters . . . . .	34
Table 100.	HCI_LE_SET_EXTENDED_ADVERTISING_PARAMETERS output parameters . . . . .	35
Table 101.	HCI_LE_SET_EXTENDED_ADVERTISING_DATA input parameters . . . . .	35
Table 102.	HCI_LE_SET_EXTENDED_ADVERTISING_DATA output parameters . . . . .	36
Table 103.	HCI_LE_SET_EXTENDED_SCAN_RESPONSE_DATA input parameters . . . . .	36
Table 104.	HCI_LE_SET_EXTENDED_SCAN_RESPONSE_DATA output parameters . . . . .	36
Table 105.	HCI_LE_SET_EXTENDED_ADVERTISING_ENABLE input parameters . . . . .	37
Table 106.	HCI_LE_SET_EXTENDED_ADVERTISING_ENABLE output parameters . . . . .	37
Table 107.	HCI_LE_READ_MAXIMUM_ADVERTISING_DATA_LENGTH output parameters . . . . .	37
Table 108.	HCI_LE_READ_NUMBER_OF_SUPPORTED_ADVERTISING_SETS output parameters . . . . .	38

Table 109. HCI_LE_REMOVE_ADVERTISING_SET input parameters . . . . .	38
Table 110. HCI_LE_REMOVE_ADVERTISING_SET output parameters . . . . .	38
Table 111. HCI_LE_CLEAR_ADVERTISING_SETS output parameters . . . . .	38
Table 112. HCI_LE_SET_EXTENDED_SCAN_PARAMETERS input parameters . . . . .	38
Table 113. HCI_LE_SET_EXTENDED_SCAN_PARAMETERS output parameters . . . . .	39
Table 114. HCI_LE_SET_EXTENDED_SCAN_ENABLE input parameters . . . . .	39
Table 115. HCI_LE_SET_EXTENDED_SCAN_ENABLE output parameters . . . . .	40
Table 116. HCI_LE_EXTENDED_CREATE_CONNECTION input parameters . . . . .	40
Table 117. HCI_LE_EXTENDED_CREATE_CONNECTION output parameters . . . . .	41
Table 118. HCI_LE_READ_TRANSMIT_POWER output parameters . . . . .	41
Table 119. HCI_LE_READ_RF_PATH_COMPENSATION output parameters . . . . .	42
Table 120. HCI_LE_WRITE_RF_PATH_COMPENSATION input parameters . . . . .	42
Table 121. HCI_LE_WRITE_RF_PATH_COMPENSATION output parameters . . . . .	42
Table 122. HCI_LE_SET_PRIVACY_MODE input parameters . . . . .	42
Table 123. HCI_LE_SET_PRIVACY_MODE output parameters . . . . .	42
Table 124. HCI_LE_GENERATE_DHKEY_V2 input parameters . . . . .	43
Table 125. HCI_LE_GENERATE_DHKEY_V2 output parameters . . . . .	43
Table 126. HCI testing commands list. . . . .	44
Table 127. HCI_LE_RECEIVER_TEST input parameters . . . . .	44
Table 128. HCI_LE_RECEIVER_TEST output parameters . . . . .	44
Table 129. HCI_LE_TRANSMITTER_TEST input parameters . . . . .	44
Table 130. HCI_LE_TRANSMITTER_TEST output parameters . . . . .	45
Table 131. HCI_LE_TEST_END output parameters . . . . .	45
Table 132. HCI_LE_RECEIVER_TEST_V2 input parameters . . . . .	45
Table 133. HCI_LE_RECEIVER_TEST_V2 output parameters . . . . .	45
Table 134. HCI_LE_TRANSMITTER_TEST_V2 input parameters . . . . .	46
Table 135. HCI_LE_TRANSMITTER_TEST_V2 output parameters . . . . .	46
Table 136. HAL commands list . . . . .	47
Table 137. ACI_HAL_GET_FW_BUILD_NUMBER output parameters . . . . .	47
Table 138. ACI_HAL_WRITE_CONFIG_DATA input parameters . . . . .	48
Table 139. ACI_HAL_WRITE_CONFIG_DATA output parameters . . . . .	48
Table 140. ACI_HAL_READ_CONFIG_DATA input parameters . . . . .	48
Table 141. ACI_HAL_READ_CONFIG_DATA output parameters . . . . .	49
Table 142. ACI_HAL_SET_TX_POWER_LEVEL input parameters . . . . .	49
Table 143. ACI_HAL_SET_TX_POWER_LEVEL output parameters . . . . .	49
Table 144. ACI_HAL_LE_TX_TEST_PACKET_NUMBER output parameters. . . . .	49
Table 145. ACI_HAL_TONE_START input parameters . . . . .	50
Table 146. ACI_HAL_TONE_START output parameters . . . . .	50
Table 147. ACI_HAL_TONE_STOP output parameters. . . . .	50
Table 148. ACI_HAL_GET_LINK_STATUS output parameters . . . . .	51
Table 149. ACI_HAL_SET_RADIO_ACTIVITY_MASK input parameters . . . . .	51
Table 150. ACI_HAL_SET_RADIO_ACTIVITY_MASK output parameters . . . . .	51
Table 151. ACI_HAL_GET_ANCHOR_PERIOD output parameters . . . . .	52
Table 152. ACI_HAL_SET_EVENT_MASK input parameters. . . . .	52
Table 153. ACI_HAL_SET_EVENT_MASK output parameters . . . . .	52
Table 154. ACI_HAL_GET_PM_DEBUG_INFO output parameters . . . . .	52
Table 155. ACI_HAL_SET_PERIPHERAL_LATENCY input parameters . . . . .	53
Table 156. ACI_HAL_SET_PERIPHERAL_LATENCY output parameters . . . . .	53
Table 157. ACI_HAL_READ_RSSI output parameters . . . . .	53
Table 158. ACI_HAL_READ_RADIO_REG input parameters. . . . .	53
Table 159. ACI_HAL_READ_RADIO_REG output parameters . . . . .	53
Table 160. ACI_HAL_WRITE_RADIO_REG input parameters . . . . .	54
Table 161. ACI_HAL_WRITE_RADIO_REG output parameters . . . . .	54
Table 162. ACI_HAL_READ_RAW_RSSI output parameters. . . . .	54
Table 163. ACI_HAL_RX_START input parameters . . . . .	54

Table 164. ACI_HAL_RX_START output parameters . . . . .	54
Table 165. ACI_HAL_RX_STOP output parameters . . . . .	55
Table 166. ACI_HAL_STACK_RESET output parameters . . . . .	55
Table 167. GAP commands list . . . . .	56
Table 168. ACI_GAP_SET_NON_DISCOVERABLE output parameters . . . . .	57
Table 169. ACI_GAP_SET_LIMITED_DISCOVERABLE input parameters . . . . .	58
Table 170. ACI_GAP_SET_LIMITED_DISCOVERABLE output parameters . . . . .	59
Table 171. ACI_GAP_SET_DISCOVERABLE input parameters. . . . .	59
Table 172. ACI_GAP_SET_DISCOVERABLE output parameters. . . . .	60
Table 173. ACI_GAP_SET_DIRECT_CONNECTABLE input parameters . . . . .	60
Table 174. ACI_GAP_SET_DIRECT_CONNECTABLE output parameters . . . . .	61
Table 175. ACI_GAP_SET_IO_CAPABILITY input parameters . . . . .	61
Table 176. ACI_GAP_SET_IO_CAPABILITY output parameters . . . . .	61
Table 177. ACI_GAP_SET_AUTHENTICATION_REQUIREMENT input parameters . . . . .	62
Table 178. ACI_GAP_SET_AUTHENTICATION_REQUIREMENT output parameters. . . . .	62
Table 179. ACI_GAP_SET_AUTHORIZATION_REQUIREMENT input parameters . . . . .	63
Table 180. ACI_GAP_SET_AUTHORIZATION_REQUIREMENT output parameters . . . . .	63
Table 181. ACI_GAP_PASS_KEY_RESP input parameters . . . . .	63
Table 182. ACI_GAP_PASS_KEY_RESP output parameters. . . . .	63
Table 183. ACI_GAP_AUTHORIZATION_RESP input parameters . . . . .	63
Table 184. ACI_GAP_AUTHORIZATION_RESP output parameters . . . . .	63
Table 185. ACI_GAP_INIT input parameters . . . . .	64
Table 186. ACI_GAP_INIT output parameters . . . . .	64
Table 187. ACI_GAP_SET_NON_CONNECTABLE input parameters. . . . .	64
Table 188. ACI_GAP_SET_NON_CONNECTABLE output parameters. . . . .	65
Table 189. ACI_GAP_SET_UNDIRECTED_CONNECTABLE input parameters . . . . .	65
Table 190. ACI_GAP_SET_UNDIRECTED_CONNECTABLE output parameters . . . . .	65
Table 191. ACI_GAP_PERIPHERAL_SECURITY_REQ input parameters. . . . .	65
Table 192. ACI_GAP_PERIPHERAL_SECURITY_REQ output parameters. . . . .	65
Table 193. ACI_GAP_UPDATE_ADV_DATA input parameters. . . . .	66
Table 194. ACI_GAP_UPDATE_ADV_DATA output parameters. . . . .	66
Table 195. ACI_GAP_DELETE_AD_TYPE input parameters. . . . .	66
Table 196. ACI_GAP_DELETE_AD_TYPE output parameters . . . . .	66
Table 197. ACI_GAP_GET_SECURITY_LEVEL input parameters . . . . .	66
Table 198. ACI_GAP_GET_SECURITY_LEVEL output parameters . . . . .	67
Table 199. ACI_GAP_SET_EVENT_MASK input parameters . . . . .	67
Table 200. ACI_GAP_SET_EVENT_MASK output parameters . . . . .	67
Table 201. ACI_GAP_CONFIGURE_FILTER_ACCEPT_LIST output parameters. . . . .	67
Table 202. ACI_GAP_TERMINATE input parameters . . . . .	68
Table 203. ACI_GAP_TERMINATE output parameters . . . . .	68
Table 204. ACI_GAP_CLEAR_SECURITY_DB output parameters. . . . .	68
Table 205. ACI_GAP_ALLOW_REBOND input parameters. . . . .	69
Table 206. ACI_GAP_ALLOW_REBOND output parameters . . . . .	69
Table 207. ACI_GAP_START_LIMITED_DISCOVERY_PROC input parameters . . . . .	69
Table 208. ACI_GAP_START_LIMITED_DISCOVERY_PROC output parameters . . . . .	69
Table 209. ACI_GAP_START_GENERAL_DISCOVERY_PROC input parameters . . . . .	70
Table 210. ACI_GAP_START_GENERAL_DISCOVERY_PROC output parameters . . . . .	70
Table 211. ACI_GAP_START_AUTO_CONNECTION_ESTABLISH_PROC input parameters . . . . .	70
Table 212. ACI_GAP_START_AUTO_CONNECTION_ESTABLISH_PROC output parameters . . . . .	71
Table 213. ACI_GAP_START_GENERAL_CONNECTION_ESTABLISH_PROC input parameters . . . . .	72
Table 214. ACI_GAP_START_GENERAL_CONNECTION_ESTABLISH_PROC output parameters . . . . .	73
Table 215. ACI_GAP_START_SELECTIVE_CONNECTION_ESTABLISH_PROC input parameters. . . . .	73
Table 216. ACI_GAP_START_SELECTIVE_CONNECTION_ESTABLISH_PROC output parameters. . . . .	74
Table 217. ACI_GAP_CREATE_CONNECTION input parameters . . . . .	74
Table 218. ACI_GAP_CREATE_CONNECTION output parameters . . . . .	75

Table 219. ACI_GAP_TERMINATE_GAP_PROC input parameters . . . . .	76
Table 220. ACI_GAP_TERMINATE_GAP_PROC output parameters . . . . .	76
Table 221. ACI_GAP_START_CONNECTION_UPDATE input parameters . . . . .	76
Table 222. ACI_GAP_START_CONNECTION_UPDATE output parameters . . . . .	76
Table 223. ACI_GAP_SEND_PAIRING_REQ input parameters . . . . .	77
Table 224. ACI_GAP_SEND_PAIRING_REQ output parameters . . . . .	77
Table 225. ACI_GAP_RESOLVE_PRIVATE_ADDR input parameters . . . . .	77
Table 226. ACI_GAP_RESOLVE_PRIVATE_ADDR output parameters . . . . .	77
Table 227. ACI_GAP_SET_BROADCAST_MODE input parameters . . . . .	78
Table 228. ACI_GAP_SET_BROADCAST_MODE output parameters . . . . .	78
Table 229. ACI_GAP_START_OBSERVATION_PROC input parameters . . . . .	79
Table 230. ACI_GAP_START_OBSERVATION_PROC output parameters . . . . .	79
Table 231. ACI_GAP_GET_BONDED_DEVICES output parameters . . . . .	80
Table 232. ACI_GAP_IS_DEVICE_BONDED input parameters . . . . .	80
Table 233. ACI_GAP_IS_DEVICE_BONDED output parameters . . . . .	80
Table 234. ACI_GAP_NUMERIC_COMPARISON_VALUE_CONFIRM_YESNO input parameters . . . . .	80
Table 235. ACI_GAP_NUMERIC_COMPARISON_VALUE_CONFIRM_YESNO output parameters . . . . .	81
Table 236. ACI_GAP_PASSKEY_INPUT input parameters . . . . .	81
Table 237. ACI_GAP_PASSKEY_INPUT output parameters . . . . .	81
Table 238. ACI_GAP_GET_OOB_DATA input parameters . . . . .	81
Table 239. ACI_GAP_GET_OOB_DATA output parameters . . . . .	81
Table 240. ACI_GAP_SET_OOB_DATA input parameters . . . . .	82
Table 241. ACI_GAP_SET_OOB_DATA output parameters . . . . .	82
Table 242. ACI_GAP_ADD_DEVICES_TO_RESOLVING_LIST input parameters . . . . .	82
Table 243. ACI_GAP_ADD_DEVICES_TO_RESOLVING_LIST output parameters . . . . .	83
Table 244. ACI_GAP_REMOVE_BONDED_DEVICE input parameters . . . . .	83
Table 245. ACI_GAP_REMOVE_BONDED_DEVICE output parameters . . . . .	83
Table 246. ACI_GAP_ADD_DEVICES_TO_LIST input parameters . . . . .	83
Table 247. ACI_GAP_ADD_DEVICES_TO_LIST output parameters . . . . .	83
Table 248. ACI_GAP_ADDITIONAL_BEACON_START input parameters . . . . .	84
Table 249. ACI_GAP_ADDITIONAL_BEACON_START output parameters . . . . .	84
Table 250. ACI_GAP_REMOVE_BONDED_DEVICE output parameters . . . . .	85
Table 251. ACI_GAP_ADDITIONAL_BEACON_SET_DATA input parameters . . . . .	85
Table 252. ACI_GAP_ADDITIONAL_BEACON_SET_DATA output parameters . . . . .	85
Table 253. ACI_GAP_ADV_SET_CONFIGURATION input parameters . . . . .	86
Table 254. ACI_GAP_ADV_SET_CONFIGURATION output parameters . . . . .	87
Table 255. ACI_GAP_ADV_SET_ENABLE input parameters . . . . .	87
Table 256. ACI_GAP_ADV_SET_ENABLE output parameters . . . . .	87
Table 257. ACI_GAP_ADV_SET_ADV_DATA input parameters . . . . .	87
Table 258. ACI_GAP_ADV_SET_ADV_DATA output parameters . . . . .	88
Table 259. ACI_GAP_ADV_SET_SCAN_RESP_DATA input parameters . . . . .	88
Table 260. ACI_GAP_ADV_SET_SCAN_RESP_DATA output parameters . . . . .	89
Table 261. ACI_GAP_ADV_REMOVE_SET input parameters . . . . .	89
Table 262. ACI_GAP_ADV_REMOVE_SET output parameters . . . . .	89
Table 263. ACI_GAP_ADV_CLEAR_SETS output parameters . . . . .	89
Table 264. ACI_GAP_ADV_SET_RANDOM_ADDRESS input parameters . . . . .	90
Table 265. ACI_GAP_ADV_SET_RANDOM_ADDRESS output parameters . . . . .	90
Table 266. ACI_GAP_EXT_START_SCAN input parameters . . . . .	90
Table 267. ACI_GAP_EXT_START_SCAN output parameters . . . . .	92
Table 268. ACI_GAP_EXT_CREATE_CONNECTION input parameters . . . . .	92
Table 269. ACI_GAP_EXT_CREATE_CONNECTION output parameters . . . . .	93
Table 270. GATT/ATT commands list . . . . .	94
Table 271. ACI_GATT_INIToutput parameters . . . . .	95
Table 272. ACI_GATT_ADD_SERVICE input parameters . . . . .	95
Table 273. ACI_GATT_ADD_SERVICE output parameters . . . . .	96

<b>Table 274.</b> ACI_GATT_INCLUDE_SERVICE input parameters . . . . .	96
<b>Table 275.</b> ACI_GATT_INCLUDE_SERVICE output parameters . . . . .	96
<b>Table 276.</b> ACI_GATT_ADD_CHAR input parameters . . . . .	97
<b>Table 277.</b> ACI_GATT_ADD_CHAR output parameters . . . . .	98
<b>Table 278.</b> ACI_GATT_ADD_CHAR_DESC input parameters . . . . .	98
<b>Table 279.</b> ACI_GATT_ADD_CHAR_DESC output parameters . . . . .	99
<b>Table 280.</b> ACI_GATT_UPDATE_CHAR_VALUE input parameters . . . . .	100
<b>Table 281.</b> ACI_GATT_UPDATE_CHAR_VALUE output parameters . . . . .	100
<b>Table 282.</b> ACI_GATT_DEL_CHAR input parameters . . . . .	100
<b>Table 283.</b> ACI_GATT_DEL_CHAR output parameters . . . . .	100
<b>Table 284.</b> ACI_GATT_DEL_SERVICE input parameters . . . . .	100
<b>Table 285.</b> ACI_GATT_DEL_SERVICE output parameters . . . . .	101
<b>Table 286.</b> ACI_GATT_DEL_INCLUDE_SERVICE input parameters . . . . .	101
<b>Table 287.</b> ACI_GATT_DEL_INCLUDE_SERVICE output parameters . . . . .	101
<b>Table 288.</b> ACI_GATT_SET_EVENT_MASK input parameters . . . . .	101
<b>Table 289.</b> ACI_GATT_SET_EVENT_MASK output parameters . . . . .	102
<b>Table 290.</b> ACI_GATT_EXCHANGE_CONFIG input parameters . . . . .	102
<b>Table 291.</b> ACI_GATT_EXCHANGE_CONFIG output parameters . . . . .	102
<b>Table 292.</b> ACI_ATT_FIND_INFO_REQ input parameters . . . . .	102
<b>Table 293.</b> ACI_ATT_FIND_INFO_REQ output parameters . . . . .	102
<b>Table 294.</b> ACI_ATT_FIND_BY_TYPE_VALUE_REQ input parameters . . . . .	103
<b>Table 295.</b> ACI_ATT_FIND_BY_TYPE_VALUE_REQ output parameters . . . . .	103
<b>Table 296.</b> ACI_ATT_READ_BY_TYPE_REQ input parameters . . . . .	103
<b>Table 297.</b> ACI_ATT_READ_BY_TYPE_REQ output parameters . . . . .	103
<b>Table 298.</b> ACI_ATT_READ_BY_GROUP_TYPE_REQ input parameters . . . . .	104
<b>Table 299.</b> ACI_ATT_READ_BY_GROUP_TYPE_REQ output parameters . . . . .	104
<b>Table 300.</b> ACI_ATT_PREPARE_WRITE_REQ input parameters . . . . .	104
<b>Table 301.</b> ACI_ATT_PREPARE_WRITE_REQ output parameters . . . . .	104
<b>Table 302.</b> ACI_ATT_EXECUTE_WRITE_REQ input parameters . . . . .	105
<b>Table 303.</b> ACI_ATT_EXECUTE_WRITE_REQ output parameters . . . . .	105
<b>Table 304.</b> ACI_GATT_DISC_ALL_PRIMARY_SERVICES input parameters . . . . .	105
<b>Table 305.</b> ACI_GATT_DISC_ALL_PRIMARY_SERVICES output parameters . . . . .	105
<b>Table 306.</b> ACI_GATT_DISC_PRIMARY_SERVICE_BY_UUID input parameters . . . . .	106
<b>Table 307.</b> ACI_GATT_DISC_PRIMARY_SERVICE_BY_UUID output parameters . . . . .	106
<b>Table 308.</b> ACI_GATT_FIND_INCLUDED_SERVICES input parameters . . . . .	106
<b>Table 309.</b> ACI_GATT_FIND_INCLUDED_SERVICES output parameters . . . . .	106
<b>Table 310.</b> ACI_GATT_DISC_ALL_CHAR_OF_SERVICE input parameters . . . . .	107
<b>Table 311.</b> ACI_GATT_DISC_ALL_CHAR_OF_SERVICE output parameters . . . . .	107
<b>Table 312.</b> ACI_GATT_DISC_CHAR_BY_UUID input parameters . . . . .	107
<b>Table 313.</b> ACI_GATT_DISC_CHAR_BY_UUID output parameters . . . . .	108
<b>Table 314.</b> ACI_GATT_DISC_ALL_CHAR_DESC input parameters . . . . .	108
<b>Table 315.</b> ACI_GATT_DISC_ALL_CHAR_DESC output parameters . . . . .	108
<b>Table 316.</b> ACI_GATT_READ_CHAR_VALUE input parameters . . . . .	109
<b>Table 317.</b> ACI_GATT_READ_CHAR_VALUE output parameters . . . . .	109
<b>Table 318.</b> ACI_GATT_READ_USING_CHAR_UUID input parameters . . . . .	109
<b>Table 319.</b> ACI_GATT_READ_USING_CHAR_UUID output parameters . . . . .	109
<b>Table 320.</b> ACI_GATT_READ_LONG_CHAR_VALUE input parameters . . . . .	110
<b>Table 321.</b> ACI_GATT_READ_LONG_CHAR_VALUE output parameters . . . . .	110
<b>Table 322.</b> ACI_GATT_READ_MULTIPLE_CHAR_VALUE input parameters . . . . .	110
<b>Table 323.</b> ACI_GATT_READ_MULTIPLE_CHAR_VALUE output parameters . . . . .	111
<b>Table 324.</b> ACI_GATT_WRITE_CHAR_VALUE input parameters . . . . .	111
<b>Table 325.</b> ACI_GATT_WRITE_CHAR_VALUE output parameters . . . . .	111
<b>Table 326.</b> ACI_GATT_WRITE_LONG_CHAR_VALUE input parameters . . . . .	112
<b>Table 327.</b> ACI_GATT_WRITE_LONG_CHAR_VALUE output parameters . . . . .	112
<b>Table 328.</b> ACI_GATT_WRITE_CHAR_RELIABLE input parameters . . . . .	112

Table 329. ACI_GATT_WRITE_CHAR_RELIABLE output parameters . . . . .	112
Table 330. ACI_GATT_WRITE_LONG_CHAR_DESC input parameters . . . . .	113
Table 331. ACI_GATT_WRITE_LONG_CHAR_DESC output parameters . . . . .	113
Table 332. ACI_GATT_READ_LONG_CHAR_DESC input parameters . . . . .	113
Table 333. ACI_GATT_READ_LONG_CHAR_DESC output parameters . . . . .	114
Table 334. ACI_GATT_WRITE_CHAR_DESC input parameters . . . . .	114
Table 335. ACI_GATT_WRITE_CHAR_DESC output parameters . . . . .	114
Table 336. ACI_GATT_READ_CHAR_DESC input parameters . . . . .	114
Table 337. ACI_GATT_READ_CHAR_DESC output parameters . . . . .	115
Table 338. ACI_GATT_WRITE_WITHOUT_RESP input parameters . . . . .	115
Table 339. ACI_GATT_WRITE_WITHOUT_RESP output parameters . . . . .	115
Table 340. ACI_GATT_SIGNED_WRITE_WITHOUT_RESP input parameters . . . . .	116
Table 341. ACI_GATT_SIGNED_WRITE_WITHOUT_RESP output parameters . . . . .	116
Table 342. ACI_GATT_CONFIRM_INDICATION input parameters . . . . .	116
Table 343. ACI_GATT_CONFIRM_INDICATION output parameters . . . . .	116
Table 344. ACI_GATT_WRITE_RESP input parameters . . . . .	117
Table 345. ACI_GATT_WRITE_RESP output parameters . . . . .	117
Table 346. ACI_GATT_ALLOW_READ input parameters . . . . .	117
Table 347. ACI_GATT_ALLOW_READ output parameters . . . . .	118
Table 348. ACI_GATT_SET_SECURITY_PERMISSION input parameters . . . . .	118
Table 349. ACI_GATT_SET_SECURITY_PERMISSION output parameters . . . . .	118
Table 350. ACI_GATT_SET_DESC_VALUE input parameters . . . . .	118
Table 351. ACI_GATT_SET_DESC_VALUE output parameters . . . . .	119
Table 352. ACI_GATT_READ_HANDLE_VALUE input parameters . . . . .	119
Table 353. ACI_GATT_READ_HANDLE_VALUE output parameters . . . . .	119
Table 354. ACI_GATT_UPDATE_CHAR_VALUE_EXT input parameters . . . . .	119
Table 355. ACI_GATT_UPDATE_CHAR_VALUE_EXT output parameters . . . . .	120
Table 356. ACI_GATT_DENY_READ input parameters . . . . .	120
Table 357. ACI_GATT_DENY_READ output parameters . . . . .	121
Table 358. ACI_GATT_SET_ACCESS_PERMISSION input parameters . . . . .	121
Table 359. ACI_GATT_SET_ACCESS_PERMISSION output parameters . . . . .	121
Table 360. ACI_GATT_STORE_DB output parameters . . . . .	121
Table 361. ACI_GATT_SET_MULT_NOTIFICATION input parameters . . . . .	122
Table 362. ACI_GATT_SET_MULT_NOTIFICATION output parameters . . . . .	122
Table 363. ACI_GATT_READ_MULTIPLE_VAR_CHAR_VALUE input parameters . . . . .	122
Table 364. ACI_GATT_READ_MULTIPLE_VAR_CHAR_VALUE output parameters . . . . .	122
Table 365. L2CAP commands list . . . . .	123
Table 366. ACI_L2CAP_CONNECTION_PARAMETER_UPDATE_REQ input parameters . . . . .	123
Table 367. ACI_L2CAP_CONNECTION_PARAMETER_UPDATE_REQ output parameters . . . . .	123
Table 368. ACI_L2CAP_CONNECTION_PARAMETER_UPDATE_RESP input parameters . . . . .	124
Table 369. ACI_L2CAP_CONNECTION_PARAMETER_UPDATE_RESP output parameters . . . . .	124
Table 370. ACI_L2CAP_CO_C_CONNECT input parameters . . . . .	124
Table 371. ACI_L2CAP_CO_C_CONNECT output parameters . . . . .	125
Table 372. ACI_L2CAP_CO_C_CONNECT_CONFIRM input parameters . . . . .	125
Table 373. ACI_L2CAP_CO_C_CONNECT_CONFIRM output parameters . . . . .	125
Table 374. ACI_L2CAP_CO_C_RECONF input parameters . . . . .	125
Table 375. ACI_L2CAP_CO_C_RECONF output parameters . . . . .	126
Table 376. ACI_L2CAP_CO_C_RECONF_CONFIRM input parameters . . . . .	126
Table 377. ACI_L2CAP_CO_C_RECONF_CONFIRM output parameters . . . . .	126
Table 378. ACI_L2CAP_CO_C_DISCONNECT input parameters . . . . .	126
Table 379. ACI_L2CAP_CO_C_DISCONNECT output parameters . . . . .	126
Table 380. ACI_L2CAP_CO_C_FLOW_CONTROL input parameters . . . . .	127
Table 381. ACI_L2CAP_CO_C_FLOW_CONTROL output parameters . . . . .	127
Table 382. ACI_L2CAP_CO_C_TX_DATA input parameters . . . . .	127
Table 383. ACI_L2CAP_CO_C_TX_DATA output parameters . . . . .	127

Table 384. HCI events commands list . . . . .	128
Table 385. HCI_DISCONNECTION_COMPLETE_EVENT parameters . . . . .	128
Table 386. HCI_ENCRYPTION_CHANGE_EVENT parameters . . . . .	129
Table 387. HCI_READ_REMOTE_VERSION_INFORMATION_COMPLETE_EVENT parameters . . . . .	129
Table 388. HCI_HARDWARE_ERROR_EVENT parameters . . . . .	129
Table 389. HCI_NUMBER_OF_COMPLETED_PACKETS_EVENT parameters . . . . .	130
Table 390. HCI_ENCRYPTION_KEY_REFRESH_COMPLETE_EVENT parameters . . . . .	130
Table 391. HCI_COMMAND_COMPLETE_EVENT parameters . . . . .	130
Table 392. HCI_COMMAND_STATUS_EVENT parameters . . . . .	131
Table 393. HCI LE META events command list . . . . .	132
Table 394. HCI_LE_CONNECTION_COMPLETE_EVENT parameters . . . . .	132
Table 395. HCI_LE_ADVERTISING_REPORT_EVENT parameters . . . . .	133
Table 396. HCI_LE_CONNECTION_UPDATE_COMPLETE_EVENT parameters . . . . .	134
Table 397. HCI_LE_READ_REMOTE_FEATURES_COMPLETE_EVENT parameters . . . . .	134
Table 398. HCI_LE_LONG_TERM_KEY_REQUEST_EVENT parameters . . . . .	134
Table 399. HCI_LE_DATA_LENGTH_CHANGE_EVENT parameters . . . . .	135
Table 400. HCI_LE_READ_LOCAL_P256_PUBLIC_KEY_COMPLETE_EVENT parameters . . . . .	135
Table 401. HCI_LE_GENERATE_DHKEY_COMPLETE_EVENT parameters . . . . .	135
Table 402. HCI_LE_ENHANCED_CONNECTION_COMPLETE_EVENT parameters . . . . .	135
Table 403. HCI_LE_DIRECTED_ADVERTISING_REPORT_EVENT parameters . . . . .	137
Table 404. HCI_LE_PHY_UPDATE_COMPLETE_EVENT parameters . . . . .	137
Table 405. HCI_LE_EXTENDED_ADVERTISING_REPORT_EVENT parameters . . . . .	137
Table 406. HCI_LE_ADVERTISING_SET_TERMINATED_EVENT parameters . . . . .	139
Table 407. HCI_LE_SCAN_REQUEST_RECEIVED_EVENT parameters . . . . .	139
Table 408. HCI_LE_CHANNEL_SELECTION_ALGORITHM_EVENT parameters . . . . .	139
Table 409. ACI GAP events commands list . . . . .	140
Table 410. ACI_GAP_PAIRING_COMPLETE_EVENT parameters . . . . .	140
Table 411. ACI_GAP_PASS_KEY_REQ_EVENT parameters . . . . .	141
Table 412. ACI_GAP_AUTHORIZATION_REQ_EVENT parameters . . . . .	141
Table 413. ACI_GAP_PROC_COMPLETE_EVENT parameters . . . . .	141
Table 414. ACI_GAP_ADDR_NOT_RESOLVED_EVENT parameters . . . . .	142
Table 415. ACI_GAP_NUMERIC_COMPARISON_VALUE_EVENT parameters . . . . .	142
Table 416. ACI_GAP_KEYPRESS_NOTIFICATION_EVENT parameters . . . . .	142
Table 417. ACI GATT/ATT events list . . . . .	143
Table 418. ACI_GATT_ATTRIBUTE_MODIFIED_EVENT parameters . . . . .	144
Table 419. ACI_GATT_PROC_TIMEOUT_EVENT parameters . . . . .	144
Table 420. ACI_ATT_EXCHANGE_MTU_RESP_EVENT parameters . . . . .	144
Table 421. ACI_ATT_FIND_INFO_RESP_EVENT parameters . . . . .	145
Table 422. ACI_ATT_FIND_BY_TYPE_VALUE_RESP_EVENT parameters . . . . .	145
Table 423. ACI_ATT_READ_BY_TYPE_RESP_EVENT parameters . . . . .	146
Table 424. ACI_ATT_READ_RESP_EVENT parameters . . . . .	146
Table 425. ACI_ATT_READ_BLOB_RESP_EVENT parameters . . . . .	146
Table 426. ACI_ATT_READ_MULTIPLE_RESP_EVENT parameters . . . . .	147
Table 427. ACI_ATT_READ_BY_GROUP_TYPE_RESP_EVENT parameters . . . . .	147
Table 428. ACI_ATT_PREPARE_WRITE_RESP_EVENT parameters . . . . .	148
Table 429. ACI_ATT_EXEC_WRITE_RESP_EVENT parameters . . . . .	148
Table 430. ACI_GATT_INDICATION_EVENT parameters . . . . .	148
Table 431. ACI_GATT_NOTIFICATION_EVENT parameters . . . . .	149
Table 432. ACI_GATT_PROC_COMPLETE_EVENT parameters . . . . .	149
Table 433. ACI_GATT_ERROR_RESP_EVENT parameters . . . . .	150
Table 434. ACI_GATT_DISC_READ_CHAR_BY_UUID_RESP_EVENT parameters . . . . .	150
Table 435. ACI_GATT_WRITE_PERMIT_REQ_EVENT parameters . . . . .	151
Table 436. ACI_GATT_READ_PERMIT_REQ_EVENT parameters . . . . .	151
Table 437. ACI_GATT_READ_MULTI_PERMIT_REQ_EVENT parameters . . . . .	152
Table 438. ACI_GATT_TX_POOL_AVAILABLE_EVENT parameters . . . . .	152

---

<b>Table 439.</b> ACI_GATT_SERVER_CONFIRMATION_EVENT parameters . . . . .	152
<b>Table 440.</b> ACI_GATT_PREPARE_WRITE_PERMIT_REQ_EVENT parameters . . . . .	153
<b>Table 441.</b> ACI_GATT_EATT_BEARER_EVENT parameters . . . . .	153
<b>Table 442.</b> ACI_GATT_MULT_NOTIFICATION_EVENT parameters . . . . .	153
<b>Table 443.</b> ACI_GATT_NOTIFICATION_COMPLETE_EVENT parameters . . . . .	154
<b>Table 444.</b> ACI_GATT_READ_EXT_EVENT parameters . . . . .	154
<b>Table 445.</b> ACI_GATT_INDICATION_EXT_EVENT parameters . . . . .	155
<b>Table 446.</b> ACI_GATT_NOTIFICATION_EXT_EVENT parameters . . . . .	155
<b>Table 447.</b> ACI L2CAP events commands list . . . . .	156
<b>Table 448.</b> ACI_L2CAP_CONNECTION_UPDATE_RESP_EVENT parameters . . . . .	156
<b>Table 449.</b> ACI_L2CAP_PROC_TIMEOUT_EVENT parameters . . . . .	156
<b>Table 450.</b> ACI_L2CAP_CONNECTION_UPDATE_REQ_EVENT parameters . . . . .	157
<b>Table 451.</b> ACI_L2CAP_COMMAND_REJECT_EVENT parameters . . . . .	157
<b>Table 452.</b> ACI_L2CAP_CO_C_CONNECT_EVENT parameters . . . . .	157
<b>Table 453.</b> ACI_L2CAP_CO_C_CONNECT_CONFIRM_EVENT parameters . . . . .	158
<b>Table 454.</b> ACI_L2CAP_CO_C_RECONF_EVENT parameters . . . . .	158
<b>Table 455.</b> ACI_L2CAP_CO_C_RECONF_CONFIRM_EVENT parameters . . . . .	158
<b>Table 456.</b> ACI_L2CAP_CO_C_DISCONNECT_EVENT parameters . . . . .	158
<b>Table 457.</b> ACI_L2CAP_CO_C_DISCONNECT_EVENT parameters . . . . .	159
<b>Table 458.</b> ACI_L2CAP_CO_C_RX_DATA_EVENT parameters . . . . .	159
<b>Table 459.</b> ACI HAL events commands list . . . . .	160
<b>Table 460.</b> ACI_HAL_END_OF_RADIO_ACTIVITY_EVENT parameters . . . . .	160
<b>Table 461.</b> ACI_HAL_SCAN_REQ_REPORT_EVENT parameters . . . . .	161
<b>Table 462.</b> ACI_HAL_FW_ERROR_EVENT parameters . . . . .	161
<b>Table 463.</b> Status error codes description . . . . .	162
<b>Table 464.</b> Tx power level . . . . .	164
<b>Table 465.</b> Document revision history . . . . .	165

**IMPORTANT NOTICE – READ CAREFULLY**

STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST's terms and conditions of sale in place at the time of order acknowledgment.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of purchasers' products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, refer to [www.st.com/trademarks](http://www.st.com/trademarks). All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2023 STMicroelectronics – All rights reserved