



7 HCI COMMANDS AND EVENTS

7.1 LINK CONTROL COMMANDS

The Link Control commands allow a Controller to control connections to other BR/EDR Controllers. Some Link Control commands are used only with a BR/EDR Controller whereas other Link Control commands are also used with an LE Controller.

In the BR/EDR Controller, when the Link Control commands are used, the Link Manager (LM) controls how the Bluetooth piconets and scatternets are established and maintained. These commands instruct the LM to create and modify Link Layer connections with Bluetooth remote devices, perform Inquiries of other BR/EDR Controllers in range, and other LMP commands.

In the LE Controller, Link Control commands are used to disconnect physical links.

For the Link Control commands, the OGF is defined as 0x01.

7.1.1 Inquiry command

Command	OCF	Command Parameters	Return Parameters
HCI_Inquiry	0x0001	LAP, Inquiry_Length, Num_Responses	

Description:

This command causes the BR/EDR Controller to enter Inquiry Mode. Inquiry Mode is used to discover other nearby BR/EDR Controllers. The LAP input parameter contains the LAP from which the inquiry access code shall be derived when the inquiry procedure is made. The Inquiry_Length parameter, added to Extended_Inquiry_Length (see [Section 6.42](#)), specifies the total duration of the Inquiry Mode and, when this time expires, Inquiry will be halted. The Num_Responses parameter specifies the number of responses that can be received before the Inquiry is halted. HCI_Inquiry_Result, HCI_Inquiry_Result_with_RSSI, or HCI_Extended_Inquiry_Result events will be sent to report the details of nearby BR/EDR Controllers that have responded to this inquiry. The HCI_Inquiry_Complete event is sent to report that Inquiry Mode has ended.

A device which responds during an inquiry or inquiry period should always be reported to the Host in an HCI_Inquiry_Result, HCI_Inquiry_Result_with_RSSI, or HCI_Extended_Inquiry_Result event if the device has not been reported earlier during the current inquiry or inquiry period and the device has not been filtered out using the command HCI_Set_Event_Filter. If the device



has been reported earlier during the current inquiry or inquiry period, whether it is reported again will depend on the implementation (e.g. whether earlier results have been saved in the BR/EDR Controller and in that case how many responses have been saved). It is recommended that the BR/EDR Controller tries to report a particular device only once during an inquiry or inquiry period.

Command parameters:**LAP:****Size: 3 octets**

Value	Parameter Description
0xXXXXXX	The LAP from which the inquiry access code should be derived when the inquiry procedure is made; see Assigned Numbers . Range: 0x9E8B00 to 0x9E8B3F

Inquiry_Length:**Size: 1 octet**

Value	Parameter Description
N = 0xXX	Maximum amount of time (added to Extended_Inquiry_Length) specified before the Inquiry is halted. Range: 0x01 to 0x30 Time = N * 1.28 s Range: 1.28 to 61.44 s

Num_Responses:**Size: 1 octet**

Value	Parameter Description
0x00	Unlimited number of responses.
0xXX	Maximum number of responses from the Inquiry before the Inquiry is halted. Range: 0x01 to 0xFF

Return parameters:

None.

Event(s) generated (unless masked away):

An HCI_Command_Status event shall be sent from the BR/EDR Controller to the Host when the BR/EDR Controller has started the Inquiry process. Unless filtered, an HCI_Inquiry_Result, HCI_Inquiry_Result_with_RSSI, or HCI_Extended_Inquiry_Result event shall be created for each BR/EDR Controller which responds to the Inquiry message. In addition, multiple BR/EDR Controllers which respond to the Inquire message may be combined into the same event. An HCI_Inquiry_Complete event shall be generated when the Inquiry process has completed.



7.1.2 Inquiry Cancel command

Command	OCF	Command Parameters	Return Parameters
HCI_Inquiry_Cancel	0x0002		Status

Description:

This command shall cause the BR/EDR Controller to stop the current Inquiry if the BR/EDR Controller is in Inquiry Mode. This command allows the Host to interrupt the BR/EDR Controller and request the BR/EDR Controller to perform a different task. The command should only be issued after the HCI_Inquiry command has been issued, an HCI_Command_Status event has been received for the Inquiry command, and before the HCI_Inquiry_Complete event occurs.

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Inquiry_Cancel command succeeded.
0x01 to 0xFF	HCI_Inquiry_Cancel command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

When the HCI_Inquiry_Cancel command has completed, an HCI_Command_Complete event shall be generated. No HCI_Inquiry_Complete event will be generated for the cancelled Inquiry process.



7.1.3 Periodic Inquiry Mode command

Command	OCF	Command Parameters	Return Parameters
HCI_Periodic_Inquiry_Mode	0x0003	Max_Period_Length, Min_Period_Length, LAP, Inquiry_Length, Num_Responses	Status

Description:

The HCI_Periodic_Inquiry_Mode command is used to configure the BR/EDR Controller to enter the Periodic Inquiry Mode that performs an automatic Inquiry. Max_Period_Length and Min_Period_Length define the time range between two consecutive inquiries, from the beginning of an inquiry until the start of the next inquiry. The BR/EDR Controller shall use this range to determine a new random time between two consecutive inquiries for each Inquiry. The LAP input parameter contains the LAP from which the inquiry access code shall be derived when the inquiry procedure is made. The Inquiry_Length parameter, added to Extended_Inquiry_Length (see [Section 6.42](#)), specifies the total duration of the Inquiry Mode and, when time expires, Inquiry will be halted. The Num_Responses parameter specifies the number of responses that can be received before the Inquiry is halted. This command is completed when the Inquiry process has been started by the BR/EDR Controller, and an HCI_Command_Complete event is sent from the Controller to the Host. When each of the periodic Inquiry processes are completed, the Controller will send an HCI_Inquiry_Complete event to the Host indicating that the latest periodic Inquiry process has finished. When a BR/EDR Controller responds to the Inquiry message, an HCI_Inquiry_Result, HCI_Inquiry_Result_with_RSSI, or HCI_Extended_Inquiry_Result event will occur to notify the Host of the discovery.

Max_Period_Length shall be greater than Min_Period_Length. Min_Period_Length shall be greater than (Inquiry_Length + Extended_Inquiry_Length).

A device which responds during an inquiry or inquiry period should always be reported to the Host in an HCI_Inquiry_Result, HCI_Inquiry_Result_with_RSSI, or HCI_Extended_Inquiry_Result event if the device has not been reported earlier during the current inquiry or inquiry period and the device has not been filtered out using the command HCI_Set_Event_Filter. If the device has been reported earlier during the current inquiry or inquiry period, whether it is reported again will depend on the implementation (e.g. whether earlier results have been saved in the BR/EDR Controller and in that case how many responses have been saved). It is recommended that the BR/EDR Controller tries to report a particular device only once during an inquiry or inquiry period.

**Command parameters:***Max_Period_Length:**Size: 2 octets*

Value	Parameter Description
N = 0xFFFF	Maximum amount of time specified between consecutive inquiries. Range: 0x0003 to 0xFFFF Time = N * 1.28 s Range: 3.84 to 83884.8 s 0.0 to 23.3 hours

*Min_Period_Length:**Size: 2 octets*

Value	Parameter Description
N = 0xFFFF	Minimum amount of time specified between consecutive inquiries. Range: 0x0002 to 0xFFFE Time = N * 1.28 s Range: 2.56 to 83883.52 s 0.0 to 23.3 hours

*LAP:**Size: 3 octets*

Value	Parameter Description
0xFFFFFFFF	The LAP from which the inquiry access code should be derived when the inquiry procedure is made; see Assigned Numbers . Range: 0x9E8B00 to 0x9E8B3F

*Inquiry_Length:**Size: 1 octet*

Value	Parameter Description
N = 0xFF	Maximum amount of time (added to Extended_Inquiry_Length) specified before the Inquiry is halted. Range: 0x01 to 0xFF Time = N * 1.28 s Range: 1.28 to 61.44 s

*Num_Responses:**Size: 1 octet*

Value	Parameter Description
0x00	Unlimited number of responses.
0xFF	Maximum number of responses from the Inquiry before the Inquiry is halted. Range: 0x01 to 0xFF



Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Periodic_Inquiry_Mode command succeeded.
0x01 to 0xFF	HCI_Periodic_Inquiry_Mode command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

The HCI Periodic Inquiry Mode begins when the BR/EDR Controller sends the HCI_Command_Complete event for this command to the Host. Unless filtered, an HCI_Inquiry_Result, HCI_Inquiry_Result_with_RSSI, or HCI_Extended_Inquiry_Result event shall be created for each remote device that has responded to the Inquiry message. In addition, multiple BR/EDR Controllers which response to the Inquiry message may be combined into the same event. An HCI_Inquiry_Complete event shall be generated when each of the periodic Inquiry processes has completed. No HCI_Inquiry_Complete event will be generated for the cancelled Inquiry process.



7.1.4 Exit Periodic Inquiry Mode command

Command	OCF	Command Parameters	Return Parameters
HCI_Exit_Periodic_Inquiry_Mode	0x0004		Status

Description:

The HCI_Exit_Periodic_Inquiry_Mode command is used to end the Periodic Inquiry mode when the local device is in Periodic Inquiry Mode. If the BR/EDR Controller is currently in an Inquiry process, the Inquiry process shall be stopped directly and the BR/EDR Controller shall no longer perform periodic inquiries until the HCI_Periodic_Inquiry_Mode command is reissued.

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Exit_Periodic_Inquiry_Mode command succeeded.
0x01 to 0xFF	HCI_Exit_Periodic_Inquiry_Mode command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

An HCI_Command_Complete event for this command shall occur when the local device is no longer in Periodic Inquiry Mode. No HCI_Inquiry_Complete event will be generated for the cancelled Inquiry process.



7.1.5 Create Connection command

Command	OCF	Command Parameters	Return Parameters
HCI_Create_Connection	0x0005	BD_ADDR, Packet_Type, Page_Scan_Repetition_Mode, Reserved, Clock_Offset, Allow_Role_Switch	

Description:

This command causes the Link Manager to create a connection to the remote device with the BD_ADDR specified by the command parameters. This command causes the local BR/EDR Controller to begin the Page process to create a link level connection. The Link Manager will determine how the new ACL connection is established. This ACL connection is determined by the current state of the device, its piconet, and the state of the device to be connected. The Packet_Type command parameter specifies which packet types the Link Manager shall use for the ACL connection; the Host shall not specify packet types that the local Controller does not support. When sending HCI ACL Data packets the Link Manager shall only use the packet type(s) specified by the Packet_Type command parameter or the always-allowed DM1 packet type. Multiple packet types may be specified for the Packet Type parameter by performing a bit-wise OR operation of the different packet types. The Link Manager may choose which packet type to be used from the list of acceptable packet types. The Page_Scan_Repetition_Mode parameter specifies the Page Scan Repetition mode supported by the remote device with the BD_ADDR. This is the most recent version of the information that was acquired either during the inquiry process or from an HCI_Page_Scan_Repetition_Mode_Change event (see [Section 7.7.31](#)). The Clock_Offset parameter is the difference between its own clock and the clock of the remote device with BD_ADDR. Only bits 2 to 16 of the difference are used, and they are mapped to this parameter as bits 0 to 14 respectively. A Clock_Offset_Valid_Flag, located in bit 15 of the Clock_Offset parameter, is used to indicate if the Clock Offset is valid or not. A Connection_Handle for this connection is returned in the HCI_Connection_Complete event (see below). The Allow_Role_Switch parameter specifies if the local device accepts or rejects the request from the remote device to switch roles at connection setup (in the Role parameter of the HCI_Accept_Connection_Request command) (before the local Controller returns an HCI_Connection_Complete event). For a definition of the different packet types see [\[Vol 2\] Part B, Section 6.5](#).

Note: The Host should enable as many packet types as possible for the Link Manager to perform efficiently.

**Command parameters:***BD_ADDR:**Size: 6 octets*

Value	Parameter Description
0XXXXXXXXXXXXX	BD_ADDR of the Device to be connected.

*Packet_Type:**Size: 2 octets*

Bit Number	Parameter Description
1	2-DH1 shall not be used.
2	3-DH1 shall not be used.
3	Ignored; DM1 may be used whether or not this bit is set.
4	DH1 may be used.
8	2-DH3 shall not be used.
9	3-DH3 shall not be used.
10	DM3 may be used.
11	DH3 may be used.
12	2-DH5 shall not be used.
13	3-DH5 shall not be used.
14	DM5 may be used.
15	DH5 may be used.
All other bits	Reserved for future use.

*Page_Scan_Repetition_Mode:**Size: 1 octet*

Value	Parameter Description
0x00	R0
0x01	R1
0x02	R2
All other values	Reserved for future use.

**Reserved:****Size: 1 octet**

Value	Parameter Description
0x00	Reserved, shall be set to 0x00.

Clock_Offset:**Size: 2 octets**

Bit Number	Parameter Description
0-14	Bits 16-2 of CLKNPeripheral - CLK
15	Clock_Offset_Valid_Flag Invalid Clock Offset = 0 Valid Clock Offset = 1

Allow_Role_Switch:**Size: 1 octet**

Value	Parameter Description
0x00	The local device will be a Central, and will not accept a role switch requested by the remote device at the connection setup.
0x01	The local device may be a Central, or may become a Peripheral after accepting a role switch requested by the remote device at the connection setup.
All other values	Reserved for future use.

Return parameters:

None.

Event(s) generated (unless masked away):

When the BR/EDR Controller receives the HCI_Create_Connection command, the BR/EDR Controller shall send the HCI_Command_Status event to the Host. In addition, when the Link Manager determines the connection is established, the BR/EDR Controller, on both BR/EDR Controllers that form the connection, shall send an HCI_Connection_Complete event to each Host. The HCI_Connection_Complete event contains the Connection_Handle if this command is successful.



7.1.6 Disconnect command

Command	OCF	Command Parameters	Return Parameters
HCI_Disconnect	0x0006	Connection_Handle, Reason	

Description:

The HCI_Disconnect command is used to terminate an existing connection. The Connection_Handle command parameter indicates which connection is to be disconnected. The Reason command parameter indicates the reason for ending the connection and is copied into the error code field of the LMP_DETACH PDU on a BR/EDR connection or the error code field of the LL_TERMINATE_IND or LL_CIS_TERMINATE_IND PDU on an LE connection. All SCO, eSCO, and CIS connections on a physical link should be disconnected before the ACL connection on the same physical connection is disconnected. If it does not, they will be implicitly disconnected as part of the ACL disconnection.

If, on the Central, the Host issues this command before issuing the HCI_LE_Create_CIS command for the same CIS, then the Controller shall return the error code *Command Disallowed* (0x0C).

If, on the Peripheral, the Host issues this command before the Controller has generated the HCI_LE_CIS_Established event for that CIS, then the Controller shall return the error code *Command Disallowed* (0x0C).

Note: As specified in [Section 7.7.5](#), on the Central, the handle for a CIS remains valid even after disconnection and, therefore, the Host can recreate a disconnected CIS at a later point in time using the same connection handle.

Command parameters:

Connection_Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xXXXX	Connection_Handle Range: 0x0000 to 0x0EFF

Reason: *Size: 1 octet*

Value	Parameter Description
0x05, 0x13 to 0x15, 0x1A, 0x29, 0x3B	<i>Authentication Failure</i> error code (0x05), <i>Other End Terminated Connection</i> error codes (0x13 to 0x15), <i>Unsupported Remote Feature</i> error code (0x1A), <i>Pairing with Unit Key Not Supported</i> error code (0x29) and <i>Unacceptable Connection Parameters</i> error code (0x3B), see [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

**Return parameters:**

None.

Event(s) generated (unless masked away):

When the Controller receives the HCI_Disconnect command, it shall send the HCI_Command_Status event to the Host. The HCI_Disconnection_Complete event will occur at each Host when the termination of the connection has completed, and on the local Host also indicates that this command has been completed. The Reason event parameter in the event on the local Host shall be set to the value *Connection Terminated by Local Host* (0x16), while that on the remote Host shall be set to the value of the Reason command parameter. However, if the termination procedure completes because a timer expires and, therefore, the local Controller cannot determine whether or not the Reason command parameter was received by the remote Controller, the Reason event parameter on the local Host should instead be set to the value *LMP Response Timeout / LL Response Timeout* (0x22).

If this command is issued for a CIS on the Central and the CIS is successfully terminated before being established, then an HCI_LE_CIS_Established event shall also be sent for this CIS with the Status *Operation Cancelled by Host* (0x44).



7.1.7 Create Connection Cancel command

Command	OCF	Command Parameters	Return Parameters
HCI_Create_ - Connection_Cancel	0x0008	BD_ADDR	Status, BD_ADDR

Description:

This command is used to request cancellation of the ongoing connection creation process, which was started by an HCI_Create_Connection command of the local BR/EDR Controller.

Command parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0xFFFFFFFFXXXX	BD_ADDR of the HCI_Create_Connection command request that was issued before and is subject of this cancellation request

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Create_Connection_Cancel command succeeded
0x01 to 0xFF	HCI_Create_Connection_Cancel command failed. See [Vol 1] Part F, Controller Error Codes for list of error codes

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0xFFFFFFFFXXXX	BD_ADDR of the HCI_Create_Connection command that was issued before and is the subject of this cancellation request.

Event(s) generated (unless masked away):

When the HCI_Create_Connection_Cancel command has completed, an HCI_Command_Complete event shall be generated.

If the connection is already established by the Baseband, but the BR/EDR Controller has not yet sent the HCI_Connection_Complete event, then the local device shall detach the link and return an HCI_Command_Complete event with the status "Success".



If the connection is already established, and the HCI_Connection_Complete event has been sent, then the Controller shall return an HCI_Command_Complete event with the error code *Connection Already Exists* (0x0B).

If the HCI_Create_Connection_Cancel command is sent to the Controller without a preceding HCI_Create_Connection command to the same device, the BR/EDR Controller shall return an HCI_Command_Complete event with the error code *Unknown Connection Identifier* (0x02).

The HCI_Connection_Complete event for the corresponding HCI_Create_Connection command shall always be sent. The HCI_Connection_Complete event shall be sent after the HCI_Command_Complete event for the HCI_Create_Connection_Cancel command. If the cancellation was successful, the HCI_Connection_Complete event will be generated with the error code *Unknown Connection Identifier* (0x02).



7.1.8 Accept Connection Request command

Command	OCF	Command Parameters	Return Parameters
HCI_Accept_Connection_Request	0x0009	BD_ADDR, Role	

Description:

The HCI_Accept_Connection_Request command is used to accept a new incoming connection request. The HCI_Accept_Connection_Request command shall only be issued after an HCI_Connection_Request event has occurred. The HCI_Connection_Request event will return the BD_ADDR of the device which is requesting the connection. This command will cause the Link Manager to create a connection to the BR/EDR Controller, with the BD_ADDR specified by the command parameters. The Link Manager will determine how the new connection will be established. This will be determined by the current state of the device, its piconet, and the state of the device to be connected. The Role command parameter allows the Host to specify if the Link Manager shall request a role switch and become the Central for this connection. This is a preference and not a requirement. If the Role Switch fails then the connection will still be accepted, and the HCI_Role_Discovery command will reflect the current role.

The Link Manager may terminate the connection if it would be low on resources if the role switch fails. The decision to accept a connection should be completed before the connection accept timeout expires on the local Bluetooth Module.

Note: When accepting a synchronous connection request, the Role parameter is not used and will be ignored by the BR/EDR Controller.

Note: See [Section 7.3.3](#) for the behavior when the HCI_Connection_Request event is masked or the connection is auto accepted.

Command parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0xFFFFFFFFXXXX	BD_ADDR of the Device to be connected

**Role:****Size: 1 octet**

Value	Parameter Description
0x00	Become the Central for this connection. The LM will perform the role switch.
0x01	Remain the Peripheral for this connection. The LM will NOT perform the role switch.

Return parameters:

None.

Event(s) generated (unless masked away):

The HCI_Accept_Connection_Request command shall cause the HCI_Command_Status event to be sent from the BR/EDR Controller when the BR/EDR Controller begins setting up the connection. In addition, when the Link Manager determines the connection is established, the local BR/EDR Controller shall send an HCI_Connection_Complete event to its Host, and the remote Controller will send an HCI_Connection_Complete event or an HCI_Synchronous_Connection_Complete event to the Host. The HCI_Connection_Complete event contains the Connection_Handle if this command is successful.



7.1.9 Reject Connection Request command

Command	OCF	Command Parameters	Return Parameters
HCI_Reject_Connection_Request	0x000A	BD_ADDR, Reason	

Description:

The HCI_Reject_Connection_Request command is used to decline a new incoming connection request. The HCI_Reject_Connection_Request command shall only be called after an HCI_Connection_Request event has occurred. The HCI_Connection_Request event will return the BD_ADDR of the device that is requesting the connection. The Reason command parameter will be returned to the connecting device in the Status parameter of the HCI_Connection_Complete event returned to the Host of the connection device, to indicate why the connection was declined.

Command parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0XXXXXXXXXXXXX	BD_ADDR of the Device to reject the connection from.

Reason:

Size: 1 octet

Value	Parameter Description
0x0D to 0x0F	Host Reject error code. See [Vol 1] Part F, Controller Error Codes for list of error codes and descriptions.

Return parameters:

None.

Event(s) generated (unless masked away):

When the Controller receives the HCI_Reject_Connection_Request command, the Controller shall send the HCI_Command_Status event to the Host. Then, the local BR/EDR Controller will send an HCI_Connection_Complete event to its Host, and the remote device shall send an HCI_Connection_Complete event or an HCI_Synchronous_Connection_Complete event to the Host. The Status parameter of the HCI_Connection_Complete event, which is sent to the Host of the device attempting to make the connection, will contain the Reason command parameter from this command.



7.1.10 Link Key Request Reply command

Command	OCF	Command Parameters	Return Parameters
HCI_Link_Key_Request_Reply	0x000B	BD_ADDR, Link_Key	Status, BD_ADDR

Description:

The HCI_Link_Key_Request_Reply command is used to reply to an HCI_Link_Key_Request event from the Controller, and specifies the Link Key stored on the Host to be used as the link key for the connection with the other BR/EDR Controller specified by BD_ADDR. The HCI_Link_Key_Request event will be generated when the BR/EDR Controller needs a Link Key for a connection.

When the BR/EDR Controller generates an HCI_Link_Key_Request event in order for the local Link Manager to respond to the request from the remote Link Manager (as a result of an HCI_Create_Connection or HCI_Authentication_Requested command from the remote Host), the local Host shall respond with either an HCI_Link_Key_Request_Reply or HCI_Link_Key_Request_Negative_Reply command before the remote Link Manager detects LMP response timeout. (See [\[Vol 2\] Part C, Link Manager Protocol Specification](#).)

When the BR/EDR Controller supports the Secure Connections (Controller Support) feature, it shall discard the Link Key once the connection has been disconnected.

Command parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0xFFFFFFFFXXXX	BD_ADDR of the Device of which the Link Key is for.

Link_Key:

Size: 16 octets

Value	Parameter Description
0xFFFFFFFFXXXX XXXXXXXXXXXX XXXXXXXXXXXX	Link Key for the associated BD_ADDR.

**Return parameters:***Status:**Size: 1 octet*

Value	Parameter Description
0x00	HCI_Link_Key_Request_Reply command succeeded.
0x01 to 0xFF	HCI_Link_Key_Request_Reply command failed. See [Vol 1] Part F, Controller Error Codes for error codes and descriptions.

*BD_ADDR:**Size: 6 octets*

Value	Parameter Description
0XXXXXXXXX XXXX	BD_ADDR of the Device of which the Link Key request reply has completed.

Event(s) generated (unless masked away):

When the HCI_Link_Key_Request_Reply command has completed, an HCI_Command_Complete event shall be generated.



7.1.11 Link Key Request Negative Reply command

Command	OCF	Command Parameters	Return Parameters
HCI_Link_Key_ - Request_Negative_ - Reply	0x000C	BD_ADDR	Status, BD_ADDR

Description:

The HCI_Link_Key_Request_Negative_Reply command is used to reply to an HCI_Link_Key_Request event from the BR/EDR Controller if the Host does not have a stored Link Key for the connection with the other BR/EDR Controller specified by BD_ADDR. The HCI_Link_Key_Request event will be generated when the BR/EDR Controller needs a Link Key for a connection.

When the Controller generates an HCI_Link_Key_Request event in order for the local Link Manager to respond to the request from the remote Link Manager (as a result of an HCI_Create_Connection or HCI_Authentication_Requested command from the remote Host), the local Host shall respond with either an HCI_Link_Key_Request_Reply or HCI_Link_Key_Request_Negative_Reply command before the remote Link Manager detects LMP response timeout. (See [\[Vol 2\] Part C, Link Manager Protocol Specification](#).)

Command parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0xFFFFFFFFXX	BD_ADDR of the Device which the Link Key is for.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Link_Key_Request_Negative_Reply command succeeded.
0x01 to 0xFF	HCI_Link_Key_Request_Negative_Reply command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0xFFFFFFFFXXXX	BD_ADDR of the Device for which the HCI_Link_Key_Request_Negative_Reply command has completed.

**Event(s) generated (unless masked away):**

When the HCI_Link_Key_Request_Negative_Reply command has completed, an HCI_Command_Complete event shall be generated.



7.1.12 PIN Code Request Reply command

Command	OCF	Command Parameters	Return Parameters
HCI_PIN_Code_Request_Reply	0x000D	BD_ADDR, PIN_Code_Length, PIN_Code	Status, BD_ADDR

Description:

The HCI_PIN_Code_Request_Reply command is used to reply to an HCI_PIN_Code_Request event from the BR/EDR Controller, and specifies the PIN code to use for a connection. The HCI_PIN_Code_Request event will be generated when a connection with remote initiating device has requested pairing.

When the BR/EDR Controller generates an HCI_PIN_Code_Request event in order for the local Link Manager to respond to the request from the remote Link Manager (as a result of an HCI_Create_Connection or HCI_Authentication_Requested command from the remote Host), the local Host shall respond with either an HCI_PIN_Code_Request_Reply or HCI_PIN_Code_Request_Negative_Reply command before the remote Link Manager detects LMP response timeout. (See [\[Vol 2\] Part C, Link Manager Protocol Specification](#).)

Command parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0XXXXXXXXX XX	BD_ADDR of the Device which the PIN code is for.

PIN_Code_Length:

Size: 1 octet

Value	Parameter Description
0xXX	The PIN code length specifies the length, in octets, of the PIN code to be used. Range: 0x01 to 0x10

PIN_Code:

Size: 16 octets

Value	Parameter Description
0XXXXXXXXX XXXXXXXXXX XXXXXXXXXX	PIN code for the device that is to be connected. The Host should ensure that strong PIN Codes are used. PIN Codes can be up to a maximum of 128 bits. Note: The PIN_Code parameter is a string parameter. Endianness does therefore not apply to the PIN_Code parameter.

**Return parameters:***Status:**Size: 1 octet*

Value	Parameter Description
0x00	HCI_PIN_Code_Request_Reply command succeeded.
0x01 to 0xFF	HCI_PIN_Code_Request_Reply command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

*BD_ADDR:**Size: 6 octets*

Value	Parameter Description
0XXXXXXXXX XXXX	BD_ADDR of the Device for which the HCI_PIN_Code_Request_Reply command has completed.

Event(s) generated (unless masked away):

When the HCI_PIN_Code_Request_Reply command has completed, an HCI_Command_Complete event shall be generated.



7.1.13 PIN Code Request Negative Reply command

Command	OCF	Command Parameters	Return Parameters
HCI_PIN_Code_Request_Negative_Reply	0x000E	BD_ADDR	Status, BD_ADDR

Description:

The HCI_PIN_Code_Request_Negative_Reply command is used to reply to a PIN Code request event from the BR/EDR Controller when the Host cannot specify a PIN code to use for a connection. This command will cause the pair request with remote device to fail.

When the BR/EDR Controller generates an HCI_PIN_Code_Request event in order for the local Link Manager to respond to the request from the remote Link Manager (as a result of an HCI_Create_Connection or HCI_Authentication_Requested command from the remote Host), the local Host shall respond with either an HCI_PIN_Code_Request_Reply or HCI_PIN_Code_Request_Negative_Reply command before the remote Link Manager detects LMP response timeout. (See [\[Vol 2\] Part C, Link Manager Protocol Specification](#).)

Command parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0xFFFFFFFFXXXX	BD_ADDR of the Device which this command is responding to.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_PIN_Code_Request_Negative_Reply command succeeded.
0x01 to 0xFF	HCI_PIN_Code_Request_Negative_Reply command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0xFFFFFFFFXXXX	BD_ADDR of the Device for which the HCI_PIN_Code_Request_Negative_Reply command has completed.

Event(s) generated (unless masked away):

When the HCI_PIN_Code_Request_Negative_Reply command has completed, an HCI_Command_Complete event shall be generated.



7.1.14 Change Connection Packet Type command

Command	OCF	Command Parameters	Return Parameters
HCI_Change_- Connection_Packet_Type	0x000F	Connection_Handle, Packet_Type	

Description:

The HCI_Change_Connection_Packet_Type command is used to change which packet types can be used for a connection that is currently established. This allows current connections to be dynamically modified to support different types of user data. The Packet_Type command parameter specifies which packet types the Link Manager can use for the connection; the Host shall not specify packet types that the local Controller does not support. When sending HCI ACL Data packets the Link Manager shall only use the packet type(s) specified by the Packet_Type command parameter or the always-allowed DM1 packet type. The interpretation of the value for the Packet_Type command parameter will depend on the Link_Type command parameter returned in the HCI_Connection_Complete event at the connection setup. Multiple packet types may be specified for the Packet_Type command parameter by bitwise OR operation of the different packet types. For a definition of the different packet types see [\[Vol 2\] Part B, Section 6.5](#).

Note: The Host should enable as many packet types as possible for the Link Manager to perform efficiently.

Note: Use the HCI_Setup_Synchronous_Connection command to change an eSCO connection.

Command parameters:

Connection_Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xXXXX	Connection_Handle Range: 0x0000 to 0x0EFF

**Packet_Type:****Size: 2 octets****For ACL Link_Type**

Bit Number	Parameter Description
1	2-DH1 shall not be used.
2	3-DH1 shall not be used.
3	Ignored; DM1 may be used whether or not this bit is set.
4	DH1 may be used.
8	2-DH3 shall not be used.
9	3-DH3 shall not be used.
10	DM3 may be used.
11	DH3 may be used.
12	2-DH5 shall not be used.
13	3-DH5 shall not be used.
14	DM5 may be used.
15	DH5 may be used.
All other bits	Reserved for future use.

For SCO Link_Type

Bit Number	Parameter Description
5	HV1 may be used.
6	HV2 may be used.
7	HV3 may be used.
All other bits	Reserved for future use.

Return parameters:

None.

Event(s) generated (unless masked away):

When the BR/EDR Controller receives the HCI_Change_Connection_Packet_Type command, the Controller shall send the HCI_Command_Status event to the Host. In addition, when the Link Manager determines the packet type has been changed for the connection, the Controller on the local device will send an HCI_Connection_Packet_Type_Changed event to the Host. This will be done at the local side only.



7.1.15 Authentication Requested command

Command	OCF	Command Parameters	Return Parameters
HCI_Authentication_Requested	0x0011	Connection_Handle	

Description:

The HCI_Authentication_Requested command is used to try to authenticate the remote device associated with the specified Connection_Handle. On an authentication failure, the BR/EDR Controller or Link Manager shall not automatically detach the link. The Host is responsible for issuing an HCI_Disconnect command to terminate the link if the action is appropriate.

The Controller shall always perform the authentication with the remote device even if the link has already been authenticated or the Controller already has a stored link key.

Note: The Connection_Handle command parameter is used to identify the other BR/EDR Controller which forms the connection. The Connection_Handle should be a Connection_Handle for an ACL connection. The authentication will apply to all Connection_Handles with the same remote BR/EDR Controller.

Command parameters:

Connection_Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xXXXX	Connection_Handle Range: 0x0000 to 0x0EFF

Return parameters:

None.

Event(s) generated (unless masked away):

When the Controller receives the HCI_Authentication_Requested command, it shall send the HCI_Command_Status event to the Host.

If Secure Simple Pairing Mode is enabled, the HCI_Link_Key_Request event shall be generated, and Secure Simple Pairing shall be started only if the Host replies to it with the HCI_Link_Key_Request_Negative_Reply command; if the Host replies to it with the HCI_Link_Key_Request_Reply command, only the authentication procedure (see [\[Vol 2\] Part C, Section 4.2.1](#)) shall be performed and no Secure Simple Pairing shall be started.



If Secure Simple Pairing Mode is not enabled, then the BR/EDR Controller may, but should not, use an existing stored link key. If authentication fails, the HCI_PIN_Code_Request event may be generated.

Using an existing stored link key when Secure Simple Pairing mode is disabled is discouraged because it does not offer the Host a method for enhancing the security of an existing link (e.g., in the case where a profile mandating a minimum passkey length is started over a link that is already authenticated with shorter passkey than the new service requires).

The HCI_Authentication_Complete event is generated when the authentication has been completed for the connection and is the indication that this command has been completed.



7.1.16 Set Connection Encryption command

Command	OCF	Command Parameters	Return Parameters
HCI_Set_Connection_ - Encryption	0x0013	Connection_Handle, Encryption_Enable	

Description:

The HCI_Set_Connection_Encryption command is used to enable and disable the link level encryption. The Connection_Handle command parameter is used to identify the other BR/EDR Controller which forms the connection. The Connection_Handle should be a Connection_Handle for an ACL connection. The encryption setting will apply to all Connection_Handles with the same remote BR/EDR Controller. While the encryption is being changed, the Link Manager will suspend all ACL-U traffic on the connection.

If both devices support both the Secure Connections (Controller Support) and Secure Connections (Host Support) features, and Encryption_Enable is set to Turn Link Level Encryption OFF when encryption is currently enabled on the specified Connection_Handle, the Controller shall return the error code *Encryption Mode Not Acceptable* (0x25).

Command parameters:

Connection_Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xXXXX	Connection_Handle Range: 0x0000 to 0x0EFF

Encryption_Enable: *Size: 1 octet*

Value	Parameter Description
0x00	Turn Link Level Encryption OFF.
0x01	Turn Link Level Encryption ON.

Return parameters:

None.

Event(s) generated (unless masked away):

When the BR/EDR Controller receives the HCI_Set_Connection_Encryption command, the BR/EDR Controller shall send the HCI_Command_Status event to the Host. When the Link Manager has completed enabling/disabling encryption for the connection, the local BR/EDR Controller shall send an



HCI_Encryption_Change event to the Host, and the BR/EDR Controller on the remote device will also generate an HCI_Encryption_Change event.



7.1.17 Change Connection Link Key command

Command	OCF	Command Parameters	Return Parameters
HCI_Change_Connection_Link_Key	0x0015	Connection_Handle	

Description:

The HCI_Change_Connection_Link_Key command is used to force both devices of a connection associated with the Connection_Handle to generate a new link key. The link key is used for authentication and encryption of connections.

Note: The Connection_Handle command parameter is used to identify the other BR/EDR Controller forming the connection. The Connection_Handle should be a Connection_Handle for an ACL connection.

Note: The resulting link key, generated as a result of HCI_Change_Connection_Link_Key command, will be of equal link key strength to the previously used link key.

Command parameters:

Connection_Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xXXXX	Connection_Handle Range: 0x0000 to 0x0EFF

Return parameters:

None.

Event(s) generated (unless masked away):

When the Controller receives the HCI_Change_Connection_Link_Key command, the BR/EDR Controller shall send the HCI_Command_Status event to the Host. When the Link Manager has changed the Link Key for the connection, the local BR/EDR Controller shall send an HCI_Link_Key_Notification event and an HCI_Change_Connection_Link_Key_Complete event to the Host, and the remote BR/EDR Controller will also generate an HCI_Link_Key_Notification event. The HCI_Link_Key_Notification event indicates that a new connection link key is valid for the connection.



7.1.18 Link Key Selection command

Command	OCF	Command Parameters	Return Parameters
HCI_Link_Key_Selection	0x0017	Key_Flag	

Description:

The HCI_Link_Key_Selection command is used to force the device that is Central of the piconet to use the temporary link key or the semi-permanent link keys. The temporary link key is used for encryption of broadcast messages within a piconet, and the semi-permanent link keys are used for private encrypted point-to-point communication. The Key_Flag command parameter is used to indicate which Link Key (temporary link key or the semi-permanent link keys) shall be used.

The HCI_Link_Key_Selection command shall be rejected with error code *Command Disallowed* (0x0C) when all Peripherals in the piconet support AES-CCM encryption and Key_Flag is set to “Use Temporary Link Key”.

Note: When at least one Peripheral in the piconet cannot support AES-CCM encryption, encrypted broadcast packets will not be received by Peripherals where both the Controller and Host support Secure Connections. When all Peripherals in the piconet support AES-CCM encryption, broadcast packets will not be encrypted and may be received by Peripherals that have AES-CCM encryption enabled.

Command parameters:

Key_Flag:

Size: 1 octet

Value	Parameter Description
0x00	Use semi-permanent Link Keys.
0x01	Use Temporary Link Key.

Return parameters:

None.

Event(s) generated (unless masked away):

When the BR/EDR Controller receives the HCI_Link_Key_Selection command, the BR/EDR Controller shall send the HCI_Command_Status event to the Host. When the Link Manager has changed link key, the BR/EDR Controller on both the local and the remote device shall send an HCI_Link_Key_Type_Changed event to the Host. If no change is required or the command fails, only the Controller on the local device shall send the event. The Connection_Handle on the Central side shall be a Connection_Handle for one of the existing



connections to a Peripheral. On the Peripheral side, the Connection_Handle shall be a Connection_Handle to the initiating Central.

The HCI_Link_Key_Type_Changed event contains the status of this command.



7.1.19 Remote Name Request command

Command	OCF	Command Parameters	Return Parameters
HCI_Remote_Name_Request	0x0019	BD_ADDR, Page_Scan_Repetition_Mode, Reserved, Clock_Offset	

Description:

The HCI_Remote_Name_Request command is used to obtain the user-friendly name of another BR/EDR Controller. The user-friendly name is used to enable the user to distinguish one BR/EDR Controller from another. The BD_ADDR command parameter is used to identify the device for which the user-friendly name is to be obtained. The Page_Scan_Repetition_Mode parameter specifies the Page Scan Repetition mode supported by the remote device with the BD_ADDR. This is the most recent version of the information that was acquired either during the inquiry process or from an HCI_Page_Scan_Repetition_Mode_Change event (see [Section 7.7.31](#)). The Clock_Offset parameter is the difference between its own clock and the clock of the remote device with BD_ADDR. Only bits 2 to 16 of the difference are used and they are mapped to this parameter as bits 0 to 14 respectively. A Clock_Offset_Valid_Flag, located in bit 15 of the Clock_Offset command parameter, is used to indicate if the Clock Offset is valid or not.

When the HCI_Remote_Supported_Host_Features_Notification event is unmasked and when the HCI_Remote_Name_Request command initiates a connection, the Link Manager shall read the remote LMP features mask pages 0 and 1.

Note: If no connection exists between the local device and the device corresponding to the BD_ADDR, a temporary Link Layer connection will be established to obtain the LMP features and name of the remote device.

Command parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0XXXXXXXXXX XX	BD_ADDR for the device whose name is requested.

**Page_Scan_Repetition_Mode:****Size: 1 octet**

Value	Parameter Description
0x00	R0
0x01	R1
0x02	R2
All other values	Reserved for future use.

Reserved:**Size: 1 octet**

Value	Parameter Description
0x00	Reserved, shall be set to 0x00.

Clock_Offset:**Size: 2 octets**

Bit Number	Parameter Description
0 to 14	Bits 2 to 16 of CLKNPeripheral - CLK
15	Clock_Offset_Valid_Flag Invalid Clock Offset = 0 Valid Clock Offset = 1

Return parameters:

None.

Event(s) generated (unless masked away):

When the BR/EDR Controller receives the HCI_Remote_Name_Request command, the BR/EDR Controller shall send the HCI_Command_Status event to the Host. If a temporary Link Layer connection was established, then when the Link Manager has completed the LMP sequence to obtain the remote Host supported features, if present, the BR/EDR Controller on the local device shall send an HCI_Remote_Host_Supported_Features_Notification event. When the Link Manager has completed the LMP messages to obtain the remote name, the local BR/EDR Controller shall send an HCI_Remote_Name_Request_Complete event to the Host. If the remote Host supported features page is present, the HCI_Remote_Host_Supported_Features_Notification event shall be sent before the HCI_Remote_Name_Request_Complete event. If not, only the HCI_Remote_Name_Request_Complete event shall be sent.



7.1.20 Remote Name Request Cancel command

Command	OCF	Command Parameters	Return Parameters
HCI_Remote_Name_Request_Cancel	0x001A	BD_ADDR	Status, BD_ADDR

Description:

This command is used to request cancellation of the ongoing remote name request process, which was started by the HCI_Remote_Name_Request command.

Command parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0xFFFFFFFFXXXX	BD_ADDR of the HCI_Remote_Name_Request command that was issued before and that is subject of this cancellation request

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Remote_Name_Request_Cancel command succeeded
0x01 to 0xFF	HCI_Remote_Name_Request_Cancel command failed. See [Vol 1] Part F, Controller Error Codes for list of error codes

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0xFFFFFFFFXXXX	BD_ADDR of the HCI_Remote_Name_Request_Cancel command that was issued before and that was subject of this cancellation request

Event(s) generated (unless masked away):

When the HCI_Remote_Name_Request_Cancel command has completed, an HCI_Command_Complete event shall be generated.

If the HCI_Remote_Name_Request_Cancel command is sent to the BR/EDR Controller without a preceding HCI_Remote_Name_Request command to the same device, the Controller shall return an HCI_Command_Complete event with the error code *Invalid HCI Command Parameters* (0x12).



The HCI_Remote_Name_Request_Complete event for the corresponding HCI_Remote_Name_Request command shall always be sent. The HCI_Remote_Name_Request_Complete event shall be sent after the HCI_Command_Complete event for the HCI_Remote_Name_Request_Cancel command. If the cancellation was successful, the HCI_Remote_Name_Request_Complete event shall be generated with the error code *Unknown Connection Identifier* (0x02).



7.1.21 Read Remote Supported Features command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Remote_Supported_Features	0x001B	Connection_Handle	

Description:

This command requests a list of the supported features for the remote device identified by the Connection_Handle parameter. The Connection_Handle shall be a Connection_Handle for an ACL-U logical link. The HCI_Read_Remote_Supported_Features_Complete event will return a list of the LMP features. For details see [\[Vol 2\] Part C, Link Manager Protocol Specification](#).

Command parameters:

Connection_Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xXXXX	Connection_Handle Range: 0x0000 to 0x0EFF

Return parameters:

None.

Event(s) generated (unless masked away):

When the BR/EDR Controller receives the HCI_Read_Remote_Supported_Features command, the BR/EDR Controller shall send the HCI_Command_Status event to the Host. When the Link Manager has completed the LMP sequence to determine the remote features or has determined that it will be using a cached copy, the BR/EDR Controller on the local device shall send an HCI_Read_Remote_Supported_Features_Complete event to the Host. The HCI_Read_Remote_Supported_Features_Complete event contains the status of this command, and parameters describing the supported features of the remote device.



7.1.22 Read Remote Extended Features command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Remote_Extended_Features	0x001C	Connection_Handle, Page_Number	

Description:

The HCI_Read_Remote_Extended_Features command returns the requested page of the extended LMP features for the remote device identified by the specified Connection_Handle. The Connection_Handle shall be the Connection_Handle for an ACL-U logical link. This command is only available if the extended features feature is implemented by the remote device. The HCI_Read_Remote_Extended_Features_Complete event will return the requested information. For details see [\[Vol 2\] Part C, Link Manager Protocol Specification](#).

Command parameters:

Connection_Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

Page_Number:

Size: 1 octet

Value	Parameter Description
0x00	Requests the normal LMP features as returned by the HCI_Read_Remote_Supported_Features command
0x01 to 0xFF	Return the corresponding page of features

Return parameters:

None.

Event(s) generated (unless masked away):

When the BR/EDR Controller receives the HCI_Read_Remote_Extended_Features command the BR/EDR Controller shall send the HCI_Command_Status event to the Host. When the Link Manager has completed the LMP sequence to determine the remote extended features or has determined that it will be using a cached copy, the Controller on the local device shall generate an HCI_Read_Remote_Extended_Features_Complete event to the Host. The HCI_Read_Remote_Extended_Features_Complete event contains the page number and the remote features returned by the remote device.



7.1.23 Read Remote Version Information command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Remote_Version_Information	0x001D	Connection_Handle	

Description:

This command will obtain the values for the version information for the remote device identified by the Connection_Handle parameter. The Connection_Handle shall be a Connection_Handle for an ACL-U or LE-U logical link.

Command parameters:

Connection_Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xXXXX	Connection_Handle Range: 0x0000 to 0x0EFF

Return parameters:

None.

Event(s) generated (unless masked away):

When the Controller receives the HCI_Read_Remote_Version_Information command, the Controller shall send the HCI_Command_Status event to the Host. When the Link Manager or Link Layer has completed the sequence to determine the remote version information, the local Controller shall send an HCI_Read_Remote_Version_Information_Complete event to the Host. The HCI_Read_Remote_Version_Information_Complete event contains the status of this command, and parameters describing the version and subversion of the LMP or Link Layer used by the remote device.



7.1.24 Read Clock Offset command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Clock_Offset	0x001F	Connection_Handle	

Description:

Both the System Clock and the clock offset to a remote device are used to determine what hopping frequency is used by a remote device for page scan. This command allows the Host to read the clock offset of remote devices. The clock offset can be used to speed up the paging procedure when the local device tries to establish a connection to a remote device, for example, when the local Host has issued an HCI_Create_Connection or HCI_Remote_Name_Request command. The Connection_Handle shall be a Connection_Handle for an ACL-U logical link.

Command parameters:

Connection_Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xXXXX	Connection_Handle Range: 0x0000 to 0x0EFF

Return parameters:

None.

Event(s) generated (unless masked away):

When the BR/EDR Controller receives the HCI_Read_Clock_Offset command, the BR/EDR Controller shall send the HCI_Command_Status event to the Host. If this command is requested at the Central then, when the Link Manager has completed the LMP messages to obtain the Clock Offset information, the BR/EDR Controller on the local BR/EDR Controller shall send an HCI_Read_Clock_Offset_Complete event to the Host. If this command is requested at the Peripheral, the LM shall immediately send an HCI_Read_Clock_Offset_Complete event to the Host, without an exchange of LMP PDUs.



7.1.25 Read LMP Handle command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_LMP_Handle	0x0020	Connection_Handle	Status, Connection_Handle, LMP_Handle, Reserved

Description:

This command reads the current LMP Handle associated with the Connection_Handle. The Connection_Handle shall identify a SCO or eSCO connection. If the Connection_Handle is a SCO Connection_Handle, then this command shall read the LMP SCO Handle for this connection. If the Connection_Handle is an eSCO Connection_Handle, then this command shall read the LMP eSCO Handle for this connection.

Command parameters:

Connection_Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_LMP_Handle command succeeded.
0x01 to 0xFF	HCI_Read_LMP_Handle command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Connection_Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

**LMP_Handle:****Size: 1 octet**

Value	Parameter Description
0xXX	The LMP Handle is the LMP Handle that is associated with this Connection_Handle. For a synchronous handle, this would be the LMP Synchronous Handle used when negotiating the synchronous connection in the link manager.

Reserved:**Size: 4 octets**

Value	Parameter Description
0x00000000	This parameter is reserved and shall be set to zero.

Event(s) generated (unless masked away):

When the HCI_Read_LMP_Handle command has completed, an HCI_Command_Complete event shall be generated.



7.1.26 Setup Synchronous Connection command

Command	OCF	Command Parameters	Return Parameters
HCI_Setup_ Synchronous_Connection	0x0028	Connection_Handle, Transmit_Bandwidth, Receive_Bandwidth, Max_Latency, Voice_Setting, Retransmission_Effort, Packet_Type	

Description:

The HCI_Setup_Synchronous_Connection command adds a new or modifies an existing synchronous logical transport (SCO or eSCO) on a physical link depending on the Connection_Handle parameter specified. If the Connection_Handle refers to an ACL link a new synchronous logical transport will be added. If the ACL link has encryption enabled using AES-CCM and the Controller cannot establish an eSCO transport (e.g. the Host parameters restricting the packet types to only SCO packet types), the Controller shall return the error code *Connection Rejected Due to Security Reasons* (0x0E) and a SCO transport will not be established. If the Connection_Handle refers to an already existing synchronous logical transport (eSCO only) this link will be modified. The parameters are specified per connection. This synchronous connection can be used to transfer synchronous voice at 64 kb/s or transparent synchronous data.

When used to setup a new synchronous logical transport, the Connection_Handle parameter shall specify an ACL connection with which the new synchronous connection will be associated. The other parameters relate to the negotiation of the link, and may be reconfigured during the lifetime of the link. The transmit and receive bandwidth specify how much bandwidth shall be available for transmitting and for receiving data. While in many cases the receive and transmit bandwidth parameters may be equal, they may be different. The latency specifies an upper limit to the time in milliseconds between the eSCO (or SCO) instants, plus the size of the retransmission window, plus the length of the reserved synchronous slots for this logical transport. The content format specifies the settings for voice or transparent data on this connection. The retransmission effort specifies the extra resources that are allocated to this connection if a packet may need to be retransmitted. The Retransmission_Effort parameter shall be set to indicate the required behavior, or to don't care.

When used to modify an existing synchronous logical transport, the Transmit_Bandwidth, Receive_Bandwidth and Voice_Setting shall be set to the same values as were used during the initial setup. The Packet_Type, Retransmission_Effort and Max_Latency parameters may be modified.



The `Packet_Type` field is a bitmap specifying which packet types the LM shall accept in the negotiation of the link parameters. Multiple packet types are specified by bitwise OR of the packet type codes in the table. At least one packet type shall be specified for each negotiation. It is recommended to enable as many packet types as possible. The Host may enable packet types that are not supported by the local Controller.

A `Connection_Handle` for the new synchronous connection will be returned in an `HCI_Synchronous_Connection_Complete` event.

Note: The link manager may choose any combination of packet types, timing, and retransmission window sizes that satisfy the parameters given. This may be achieved by using more frequent transmissions of smaller packets. The link manager may choose to set up either a SCO or an eSCO connection, if the parameters allow, using the corresponding LMP sequences.

Note: To modify a SCO connection, use the `HCI_Change_Connection_Packet_Type` command.

If the lower layers cannot achieve the exact transmit and receive bandwidth requested subject to the other parameters, then the link shall be rejected.

A synchronous connection may only be created when an ACL connection already exists.

Command parameters:

Connection_Handle: 2 octets (12 bits meaningful)

Value	Parameter Description
0xXXXX	Connection_Handle Range: 0x0000 to 0x0EFF

Transmit_Bandwidth: 4 octets

Value	Parameter Description
0XXXXXXXX	Transmit bandwidth in octets per second.

Receive_Bandwidth: 4 octets

Value	Parameter Description
0XXXXXXXX	Receive bandwidth in octets per second.

**Max_Latency:****2 octets**

Value	Parameter Description
0x0000 to 0x0003	Reserved for future use
0x0004 to 0xFFFE	This is a value in milliseconds representing the upper limit of the sum of the synchronous interval, and the size of the eSCO window, where the eSCO window is the reserved slots plus the retransmission window. (See [Vol 2] Part B, Figure 8.9)
0xFFFF	Don't care.

Voice_Setting:**2 octets (10 bits meaningful)**

Value	Parameter Description
See Section 6.12 .	

Retransmission_Effort:**1 octet**

Value	Parameter Description
0x00	No retransmissions (SCO or eSCO connection allowed)
0x01	At least one retransmission, optimize for power consumption (eSCO connection required).
0x02	At least one retransmission, optimize for link quality (eSCO connection required)
0xFF	Don't care (SCO or eSCO connection allowed)
All other values	Reserved for future use

Packet_Type:**2 octets**

Bit Number	Parameter Description
0	HV1 may be used.
1	HV2 may be used.
2	HV3 may be used.
3	EV3 may be used.
4	EV4 may be used.
5	EV5 may be used.
6	2-EV3 shall not be used.
7	3-EV3 shall not be used.
8	2-EV5 shall not be used.
9	3-EV5 shall not be used.
All other bits	Reserved for future use

**Return parameters:**

None.

Event(s) generated (unless masked away):

When the BR/EDR Controller receives the HCI_Setup_Synchronous_Connection command, it shall send the HCI_Command_Status event to the Host. In addition, when the LM determines the connection is established, the local BR/EDR Controller shall send an HCI_Synchronous_Connection_Complete event to the local Host, and the remote Controller will send an HCI_Synchronous_Connection_Complete event or an HCI_Connection_Complete event to the remote Host. The HCI_Synchronous_Connection_Complete event contains the Connection_Handle if this command is successful.

If this command is used to change the parameters of an existing eSCO link, the HCI_Synchronous_Connection_Changed event is sent to both Hosts. In this case no HCI_Synchronous_Connection_Complete event, HCI_Connection_Request event, or HCI_Connection_Complete event will be sent to either Host. This command cannot be used to change the parameters of a SCO link.



7.1.27 Accept Synchronous Connection Request command

Command	OCF	Command Parameters	Return Parameters
HCI_Accept_Synchronous_Connection_Request	0x0029	BD_ADDR, Transmit_Bandwidth, Receive_Bandwidth, Max_Latency, Voice_Setting, Retransmission_Effort, Packet_Type	

Description:

The HCI_Accept_Synchronous_Connection_Request command is used to accept an incoming request for a synchronous connection and to inform the local Link Manager about the acceptable parameter values for the synchronous connection.

The command shall only be issued after an HCI_Connection_Request event with link type SCO or eSCO has occurred. The HCI_Connection_Request event contains the BD_ADDR of the device requesting the connection. The decision to accept a connection should be taken before the timer Connection_Accept_Timeout expires on the local device.

The Host shall include in the Packet_Type parameter at least one packet type for the transport (SCO or eSCO) specified in the incoming request. The Controller shall ignore any packet types in the Packet_Type parameter for the other transport.

If the ACL link has encryption enabled using AES-CCM then the Host shall not accept a request where the link type is SCO.

The parameter set of the HCI_Accept_Synchronous_Connection_Request command is the same as for the HCI_Setup_Synchronous_Connection command. The Transmit_Bandwidth and Receive_Bandwidth values are required values for the new link and shall be met. The Max_Latency is an upper bound to the acceptable latency for the Link, as defined in [Section 7.1.26 Setup_Synchronous_Connection](#) and shall not be exceeded. Voice_Setting specifies the encoding in the same way as in the HCI_Setup_Synchronous_Connection command and shall be met. The Retransmission_Effort parameter shall be set to indicate the required behavior, or to don't care. The Packet_Type parameter is a bit mask specifying the synchronous packet types that are allowed on the link.

If the Link Type of the incoming request is SCO, then only the Transmit_Bandwidth, Max_Latency, Voice_Setting, and Packet_Type fields are valid.



Note: See [Section 7.3.3](#) for the behavior when the HCI_Connection_Request event is masked or the connection is auto accepted.

Command parameters:

BD_ADDR: 6 octets

Value	Parameter Description
0xFFFFFFFFXXXXXX	BD_ADDR of the device requesting the connection

Transmit_Bandwidth: 4 octets

Value	Parameter Description
0x00000000 to 0xFFFFFFFF	Transmit bandwidth in octets per second.
0xFFFFFFFF	Don't care

Default: Don't care

Receive_Bandwidth: 4 octets

Value	Parameter Description
0x00000000 to 0xFFFFFFFF	Receive bandwidth in octets per second.
0xFFFFFFFF	Don't care

Default: Don't care

Max_Latency: 2 octets

Value	Parameter Description
0x0000 to 0x0003	Reserved for future use
0x0004 to 0xFFFF	This is a value in milliseconds representing the upper limit of the sum of the synchronous interval and the size of the eSCO window, where the eSCO window is the reserved slots plus the retransmission window. (See [Vol 2] Part B, Figure 8.9)
0xFFFF	Don't care.

Default: Don't care

Voice_Setting: 2 octets (10 bits meaningful)

Value	Parameter Description
	See Section 6.12 .



Default: When links are auto-accepted, the values written by the HCI_Write_Voice_Setting command are used.

Retransmission_Effort:

1 octet

Value	Parameter Description
0x00	No retransmissions (SCO or eSCO connection allowed)
0x01	At least one retransmission, optimize for power consumption (eSCO connection required).
0x02	At least one retransmission, optimize for link quality (eSCO connection required).
0xFF	Don't care (SCO or eSCO connection allowed)
All other values	Reserved for future use

Default: Don't care

Packet_Type:

2 octets

Bit Number	Parameter Description
0	HV1 may be used.
1	HV2 may be used.
2	HV3 may be used.
3	EV3 may be used.
4	EV4 may be used.
5	EV5 may be used.
6	2-EV3 shall not be used.
7	3-EV3 shall not be used
8	2-EV5 shall not be used.
9	3-EV5 shall not be used.
All other bits	Reserved for future use

Default: 0x003F - means all defined packet types may be used.

Return parameters:

None.

Event(s) generated (unless masked away):

The HCI_Accept_Synchronous_Request command shall cause the HCI_Command_Status event to be sent from the BR/EDR Controller when the BR/EDR Controller starts setting up the connection. When the link setup is complete, the local BR/EDR Controller shall send an HCI_Synchronous_



Connection_Complete event to its Host, and the remote BR/EDR Controller will send an HCI_Synchronous_Connection_Complete event to the Host. The HCI_Synchronous_Connection_Complete will contain the Connection_Handle and the link parameters if the setup is successful.



7.1.28 Reject Synchronous Connection Request command

Command	OCF	Command Parameters	Return Parameters
HCI_Reject_Synchronous_Connection_Request	0x002A	BD_ADDR, Reason	

Description:

The HCI_Reject_Synchronous_Connection_Request command is used to decline an incoming request for a synchronous link. It shall only be issued after an HCI_Connection_Request event with Link Type equal to SCO or eSCO has occurred. The HCI_Connection_Request event contains the BD_ADDR of the device requesting the connection. The Reason parameter will be returned to the initiating Host in the Status parameter of the HCI_Synchronous_Connection_Complete event on the remote side.

If the ACL link has encryption enabled using AES-CCM and the requested link type was SCO, the Host shall reject the request using this command with the Reason parameter set to *Connection Rejected Due to Security Reasons* (0x0E).

Command parameters:

BD_ADDR: 6 octets

Value	Parameter Description
0xFFFFFFFFXXXX	BD_ADDR of the device requesting the connection

Reason: 1 octet

Value	Parameter Description
0x0D to 0x0F	Host Reject error code. See [Vol 1] Part F, Controller Error Codes for error codes and descriptions.

Return parameters:

None.

Event(s) generated (unless masked away):

When the Controller receives the HCI_Reject_Synchronous_Connection_Request, it shall send an HCI_Command_Status event to the Host. When the setup is terminated, the local Controller shall send an HCI_Synchronous_Connection_Complete event to its Host, and the remote Controller will send an HCI_Synchronous_Connection_Complete event to the Host with the Reason code from this command.



7.1.29 IO Capability Request Reply command

Command	OCF	Command Parameters	Return Parameters
HCI_IO_Capability_Request_Reply	0x002B	BD_ADDR, IO_Capability, OOB_Data_Present, Authentication_Requirements	Status, BD_ADDR

Description:

The HCI_IO_Capability_Request_Reply command is used to reply to an HCI_IO_Capability_Request event from the Controller, and specifies the current I/O capabilities of the Host. This includes the Host input, output and out-of-band (OOB) capabilities.

If an authenticated link key is not required by the Host, the Authentication Requirements parameter may be set to one of the following:

- MITM Protection Not Required – No Bonding
- MITM Protection Not Required – Dedicated Bonding
- MITM Protection Not Required – General Bonding

If both Hosts set the Authentication_Requirements parameter to one of the above values, the Link Managers shall use the numeric comparison authentication procedure and the Hosts shall use the Just Works association model.

If an authenticated link key is required by the Host, the Authentication Requirements parameter shall be set to one of the following:

- MITM Protection Required – No Bonding
- MITM Protection Required – Dedicated Bonding
- MITM Protection Required – General Bonding

In addition, the following requirements apply:

- If one or both Hosts set the Authentication Requirements parameter to one of the above values, the Link Managers shall use the IO_Capability parameter to determine the authentication procedure.
- A Host that sets the Authentication_Requirements parameter to one of the above values shall verify that the resulting Link Key type meets the security requirements requested.

If the Host has received OOB authentication data from a device with the same BD_ADDR sent in the HCI_IO_Capability_Request event, then the OOB_Data_Present parameter shall be set to:

- "P-192 OOB authentication data from remote device present" when the Host has received only P-192 OOB data,



- "P-256 OOB authentication data from remote device present" when the Host has received only P-256 OOB data, or
- "P-192 and P-256 OOB authentication data from remote device present" when the Host has received both P-192 and P-256 OOB data.

Otherwise OOB_Data_Present shall be set to "OOB authentication data not present".

Command parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0XXXXXXXXXX XX	BD_ADDR of remote device involved in the Secure Simple Pairing process

IO_Capability:

Size: 1 octet

Value	Parameter Description
0x00	DisplayOnly
0x01	DisplayYesNo
0x02	KeyboardOnly
0x03	NoInputNoOutput
All other values	Reserved for future use

OOB_Data_Present:

Size: 1 octet

Value	Parameter Description
0x00	OOB authentication data not present
0x01	P-192 OOB authentication data from remote device present
0x02	P-256 OOB authentication data from remote device present
0x03	P-192 and P-256 OOB authentication data from remote device present
All other values	Reserved for future use

**Authentication_Requirements:****Size: 1 octet**

Value	Parameter Description
0x00	MITM Protection Not Required – No Bonding. Numeric comparison with automatic accept allowed.
0x01	MITM Protection Required – No Bonding. Use IO Capabilities to determine authentication procedure
0x02	MITM Protection Not Required – Dedicated Bonding. Numeric comparison with automatic accept allowed.
0x03	MITM Protection Required – Dedicated Bonding. Use IO Capabilities to determine authentication procedure
0x04	MITM Protection Not Required – General Bonding. Numeric Comparison with automatic accept allowed.
0x05	MITM Protection Required – General Bonding. Use IO capabilities to determine authentication procedure.
All other values	Reserved for future use

Return parameters:**Status:****Size: 1 octet**

Value	Parameter Description
0x00	HCI_IO_Capability_Request_Reply command succeeded
0x01 to 0xFF	HCI_IO_Capability_Request_Reply command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

BD_ADDR:**Size: 6 octets**

Value	Parameter Description
0XXXXXXXXXX XX	BD_ADDR of remote device involved in the Secure Simple Pairing process

Event(s) generated (unless masked away):

When the HCI_IO_Capability_Request_Reply command has completed, an HCI_Command_Complete event shall be generated. When the device is the initiator of Secure Simple Pairing, an HCI_IO_Capability_Response event shall be generated. Additionally, when the OOB_Data_Present parameter indicates that OOB authentication data from the remote device is present, the HCI_Remote_OOB_Data_Request event shall be generated.



7.1.30 User Confirmation Request Reply command

Command	OCF	Command Parameters	Return Parameters
HCI_User_Confirmation_Request_Reply	0x002C	BD_ADDR	Status, BD_ADDR

Description:

The HCI_User_Confirmation_Request_Reply command is used to reply to an HCI_User_Confirmation_Request event and indicates that the user selected "yes". It is also used when the Host has no input and no output capabilities.

Command parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0XXXXXXXXXX X	BD_ADDR of remote device involved in the Secure Simple Pairing process

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_User_Confirmation_Request_Reply command succeeded
0x01 to 0xFF	HCI_User_Confirmation_Request_Reply command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0XXXXXXXXXX X	BD_ADDR of remote device involved in the Secure Simple Pairing process

Event(s) generated (unless masked away):

When the HCI_User_Confirmation_Request_Reply command has completed, an HCI_Command_Complete event shall be generated.



7.1.31 User Confirmation Request Negative Reply command

Command	OCF	Command Parameters	Return Parameters
HCI_User_Confirmation_Request_Negative_Reply	0x002D	BD_ADDR	Status, BD_ADDR

Description:

The HCI_User_Confirmation_Request_Negative_Reply command is used to reply to an HCI_User_Confirmation_Request event and indicates that the user selected "no". This command shall cause the initiating Link Manager to transmit an LMP_NUMERIC_COMPARISON_FAILED PDU and terminate Secure Simple Pairing.

Command parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0XXXXXXXXXX XX	BD_ADDR of remote device involved in the Secure Simple Pairing process

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_User_Confirmation_Request_Negative_Reply command succeeded
0x01 to 0xFF	HCI_User_Confirmation_Request_Negative_Reply command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0XXXXXXXXXX XX	BD_ADDR of remote device involved in the Secure Simple Pairing process

Event(s) generated (unless masked away):

When the HCI_User_Confirmation_Request_Negative_Reply command has completed, an HCI_Command_Complete event shall be generated.



7.1.32 User Passkey Request Reply command

Command	OCF	Command Parameters	Return Parameters
HCI_User_Passkey_Request_Reply	0x002E	BD_ADDR, Numeric_Value	Status, BD_ADDR

Description:

The HCI_User_Passkey_Request_Reply command is used to reply to an HCI_User_Passkey_Request event and specifies the Numeric_Value (passkey) entered by the user to be used in the Secure Simple Pairing process.

Command parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0xFFFFFFFFXXXX	BD_ADDR of remote device involved in the Secure Simple Pairing process

Numeric_Value:

Size: 4 octets

Value	Parameter Description
0x00000000 to 0x000F423F	Numeric value (passkey) entered by user. Valid values are decimal 000000 to 999999.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_User_Passkey_Request_Reply command succeeded
0x01 to 0xFF	HCI_User_Passkey_Request_Reply command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0xFFFFFFFFXXXX	BD_ADDR of remote device involved in the Secure Simple Pairing process

Event(s) generated (unless masked away):

When the HCI_User_Passkey_Request_Reply command has completed, an HCI_Command_Complete event shall be generated.



7.1.33 User Passkey Request Negative Reply command

Command	OCF	Command Parameters	Return Parameters
HCI_User_Passkey_ - Request_Negative_Reply	0x002F	BD_ADDR	Status, BD_ADDR

Description:

The HCI_User_Passkey_Request_Negative_Reply command is used to reply to an HCI_User_Passkey_Request event and indicates the Host could not provide a passkey. This command shall cause the initiating Link Manager to transmit an LMP_PASSKEY_ENTRY_FAILED PDU and terminate Simple Pairing.

Command parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0XXXXXXXXXX XX	BD_ADDR of remote device involved in the Secure Simple Pairing process

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_User_Passkey_Request_Negative_Reply command succeeded
0x01 to 0xFF	HCI_User_Passkey_Request_Negative_Reply command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0XXXXXXXXXX XX	BD_ADDR of remote device involved in the Secure Simple Pairing process

Event(s) generated (unless masked away):

When the HCI_User_Passkey_Negative_Request_Reply command has completed, an HCI_Command_Complete event shall be generated.



7.1.34 Remote OOB Data Request Reply command

Command	OCF	Command Parameters	Return Parameters
HCI_Remote_OOB_Data_Request_Reply	0x0030	BD_ADDR, C, R	Status, BD_ADDR

Description:

The HCI_Remote_OOB_Data_Request_Reply command is used to reply to an HCI_Remote_OOB_Data_Request event with the C and R values received via an OOB transfer from a remote device identified by BD_ADDR. If the R value is not present in the received OOB data from the remote device, the Host shall set R to zeros.

Command parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0XXXXXXXXXX XX	BD_ADDR of remote device from which the C and R values were received

C:

Size: 16 octets

Value	Parameter Description
0XXXXXXXXXX XXXXXXXXXXXX XXXXXXXXXXXX	Secure Simple Pairing Hash C

R:

Size: 16 octets

Value	Parameter Description
0XXXXXXXXXX XXXXXXXXXXXX XXXXXXXXXXXX	Secure Simple Pairing Randomizer R

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Remote_OOB_Data_Request_Reply command succeeded
0x01to 0xFF	HCI_Remote_OOB_Data_Request_Reply command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

**BD_ADDR:****Size: 6 octets**

Value	Parameter Description
0XXXXXXXXXX XX	BD_ADDR of remote device involved in the Secure Simple Pairing process

Event(s) generated (unless masked away):

When the HCI_Remote_OOB_Data_Request_Reply command has completed, an HCI_Command_Complete event shall be generated.



7.1.35 Remote OOB Data Request Negative Reply command

Command	OCF	Command Parameters	Return Parameters
HCI_Remote_OOB_Data_Request_Negative_Reply	0x0033	BD_ADDR	Status, BD_ADDR

Description:

The HCI_Remote_OOB_Data_Request_Negative_Reply command is used to reply to an HCI_Remote_OOB_Data_Request event that the Host does not have the C and R values associated with the remote device identified by BD_ADDR.

Command parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0XXXXXXXXXXXXX	BD_ADDR of remote device

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Remote_OOB_Data_Request_Negative_Reply command succeeded
0x01 to 0xFF	HCI_Remote_OOB_Data_Request_Negative_Reply command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0XXXXXXXXXXXXX X	BD_ADDR of remote device involved in the Secure Simple Pairing process

Event(s) generated (unless masked away):

When the HCI_Remote_OOB_Data_Request_Negative_Reply command has completed, an HCI_Command_Complete event shall be generated.



7.1.36 IO Capability Request Negative Reply command

Command	OCF	Command Parameters	Return Parameters
HCI_IO_Capability_Request_Negative_Reply	0x0034	BD_ADDR, Reason	Status, BD_ADDR

Description:

The HCI_IO_Capability_Request_Negative_Reply command shall be used to reject a pairing attempt after an HCI_IO_Capability_Request event has been received by the Host. The reason for the rejection is given in the Reason parameter. The error code *Secure Simple Pairing not Supported by Host* (0x37) shall not be used in the Reason parameter.

Command parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0XXXXXXXXXXXXX	BD_ADDR of remote device involved in the Secure Simple Pairing process

Reason:

Size: 1 octet

Value	Parameter Description
0xXX	Reason that Secure Simple Pairing rejected. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_IO_Capability_Request_Negative_Reply command succeeded.
0x01 to 0xFF	HCI_IO_Capability_Request_Negative_Reply command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0XXXXXXXXXXXXX	BD_ADDR of remote device involved in the Secure Simple Pairing process

**Event(s) generated (unless masked away):**

When the HCI_IO_Capability_Request_Negative_Reply command has completed, an HCI_Command_Complete event shall be generated.



7.1.37 [This section is no longer used]

7.1.38 [This section is no longer used]

7.1.39 [This section is no longer used]

7.1.40 [This section is no longer used]

7.1.41 [This section is no longer used]

7.1.42 [This section is no longer used]

7.1.43 [This section is no longer used]

7.1.44 [This section is no longer used]



7.1.45 Enhanced Setup Synchronous Connection command

Command	OCF	Command Parameters	Return Parameters
HCI_Enhanced_Setup_Synchronous_Connection	0x003D	Connection_Handle, Transmit_Bandwidth, Receive_Bandwidth, Transmit_Coding_Format, Receive_Coding_Format, Transmit_Codec_Frame_Size, Receive_Codec_Frame_Size, Input_Bandwidth, Output_Bandwidth, Input_Coding_Format, Output_Coding_Format, Input_Coded_Data_Size, Output_Coded_Data_Size, Input_PCM_Data_Format, Output_PCM_Data_Format, Input_PCM_Sample_Payload_MSB_Position, Output_PCM_Sample_Payload_MSB_Position, Input_Data_Path, Output_Data_Path, Input_Transport_Unit_Size, Output_Transport_Unit_Size, Max_Latency, Packet_Type, Retransmission_Effort	

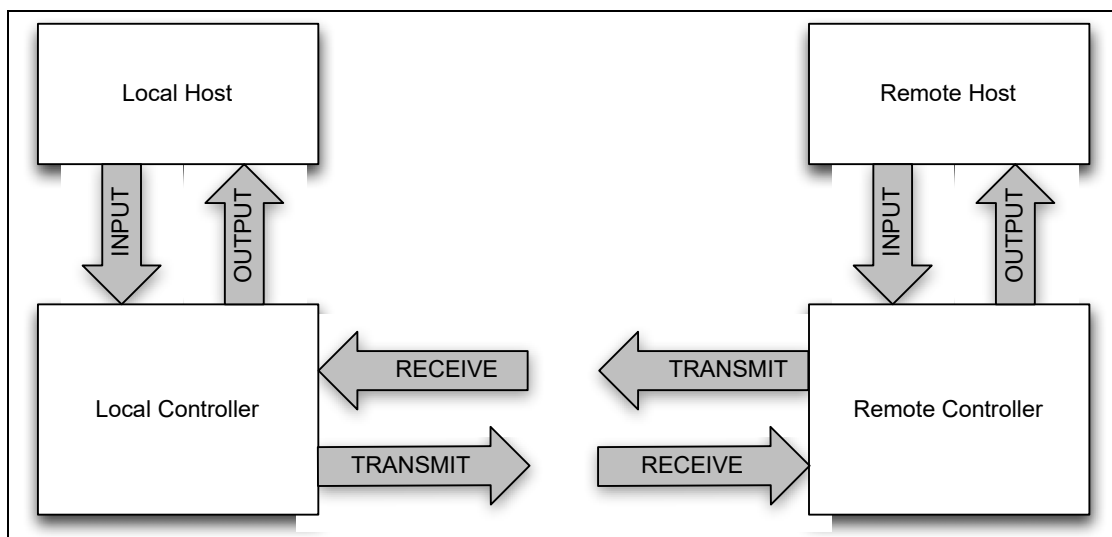
Description:

The HCI_Enhanced_Setup_Synchronous_Connection command adds a new, or modifies an existing, synchronous logical transport (SCO or eSCO) on a physical link depending on the Connection_Handle parameter specified. If the Connection_Handle refers to an ACL link, then a new synchronous logical transport shall be added. If the Connection_Handle refers to an existing synchronous logical transport (eSCO only), then this link shall be modified. The parameters are specified per connection. This synchronous connection can be used to transfer synchronous voice data or transparent synchronous data. If the ACL link has encryption enabled using AES-CCM and the Controller cannot establish an eSCO transport (e.g. the Host parameters restricting the packet types to SCO packet types), the Controller shall return the error code *Connection Rejected Due to Security Reasons* (0x0E) and a SCO transport will not be established.



When used to setup a new synchronous logical transport, the `Connection_Handle` parameter shall specify an ACL connection with which the new synchronous connection shall be associated. The other parameters relate to the negotiation of the link, and may be reconfigured during the lifetime of the link.

The following terms are used to describe the four different audio paths: Transmit, Receive, Input and Output. The Transmit and Receive paths are from the perspective of the local Controller's radio. The Input and Output paths are from the perspective of the Controller.



The following parameters are used to describe the transmit and receive paths over the air:

- The `Transmit_Bandwidth` and `Receive_Bandwidth` parameters specify how much bandwidth shall be available for transmitting and for receiving data. The Host shall set the `Transmit_Bandwidth` and `Receive_Bandwidth` parameters to be equal or shall set one of them to be zero and the other non-zero.
- The `Transmit_Coding_Format` and `Receive_Coding_Format` parameters specify the coding format used for transmitted or received data. The Host shall set the `Transmit_Coding_Format` and `Receive_Coding_Formats` to be equal. When the `Transmit_Coding_Format` and `Receive_Coding_Format` parameters are not equal to CVSD, A-law or μ -law, the Link Manager shall map these to Transparent air mode.
- The `Transmit_Codec_Frame_Size` and `Receive_Codec_Frame_Size` parameters specify the frame size produced by the codecs in the context of over-the-air coding. The over-the-air packet size should have the following relationship with the codec frame size:

$$\text{Packet_Size} = \text{Frame_Size} * N, \text{ or}$$

$$\text{Packet_Size} = \text{Frame_Size} / N$$

where N is an integer.



The following parameters are used to describe the coding format used prior to encapsulating over the audio data transport path:

- The `Input_Bandwidth` and `Output_Bandwidth` specify the nominal rate at which the Host or Controller transfers data (for HCI transports this excludes the HCI header). The Host shall either set the `Input_Bandwidth` and `Output_Bandwidth` to be equal, or shall set one of them to be zero and the other non-zero.
- The `Input_Coding_Format` and `Output_Coding_Format` parameters specify the coding format used over the transport. The Host shall set the `Input_Coding_Format` and `Output_Coding_Format` to be equal.
- The `Input_Coded_Data_Size` and `Output_Coded_Data_Size` specify the number of bits in each sample or frame of data. For CVSD, a frame of data shall be 8 bits.
- The `Input_PCM_Data_Format` and `Output_PCM_Data_Format` parameters specify the data format over the transport for linear samples. They shall be ignored when the data is encoded in any other way.
- The `Input_PCM_Sample_Payload_MSB_Position` and `Output_PCM_Sample_Payload_MSB_Position` parameters indicate, for linear samples, how many bit positions that the MSB of the sample is away from starting at the MSB of the data. They shall be ignored when the data is encoded in any other way. For example, if `Input_Coded_Data_Size` = 16 and `Input_PCM_Sample_Payload_MSB_Position` = 3, then each sample is actually only 13 bits, the MSB (which is the sign bit for signed formats) is bit 12 (counting from the LSB at bit 0), and the contents of bits 13, 14, and 15 of each sample shall be ignored.

The following parameters describe the audio data transport path characteristics:

- The `Input_Data_Path` and `Output_Data_Path` parameters specify the audio data transport path. When set to 0x00, the audio data path shall be over the HCI transport. When set to 0xFF, audio test mode (see [Section 7.6.2](#)) is selected (this is only applicable during test mode). When set to 0x01 to 0xFE, the audio data path shall use non-HCI transport data paths (e.g. PCM interface) with logical transport channel numbers. The meanings of these logical transport channel numbers are vendor specific.
- The `Input_Transport_Unit_Size` and `Output_Transport_Unit_Size` indicate how many bits are in each unit of data delivered by the audio data transport. Except for HCI, the meaning of “unit” depends on the Host transport used and, therefore, is vendor specific (for example, on a PCM transport this should indicate the number of bits transported per sync pulse, and would normally be 8 or 16). The Host shall set the `Input_Transport_Unit_Size` and `Output_Transport_Unit_Size` to be equal. For HCI Host transport the Host shall set them to 0.



The following parameters are used by the Link Manager to negotiate the synchronous transport:

- The `Max_Latency` parameter specifies an upper limit to the time in milliseconds between the eSCO (or SCO) instants, plus the size of the retransmission window, plus the length of the reserved synchronous slots for this logical transport.
- The `Packet_Type` parameter is a bitmap specifying which synchronous packet types may be used by the Link Manager in the negotiation of the link parameters. Multiple packet types are specified by bitwise OR of the packet type codes in the table. At least one packet type shall be specified for each negotiation. It is recommended to enable as many packet types as possible. The Host may enable packet types that are not supported by the local Controller.
- The `Retransmission_Effort` parameter specifies the extra resources that are allocated to this connection if a packet may need to be retransmitted. The `Retransmission_Effort` parameter shall be set to indicate the required behavior, or to “Don’t care”.

The following restrictions shall apply:

- Either both the `Transmit_Coding_Format` and `Input_Coding_Format` shall be “transparent” or neither shall be. If both are “transparent”, the `Transmit_Bandwidth` and the `Input_Bandwidth` shall be the same and the Controller shall not modify the data sent to the remote device.
- Either both the `Receive_Coding_Format` and `Output_Coding_Format` shall be “transparent” or neither shall be. If both are “transparent”, the `Receive_Bandwidth` and the `Output_Bandwidth` shall be the same and the Controller shall not modify the data sent to the Host.

A `Connection_Handle` for the new synchronous connection will be returned in the `HCI_Synchronous_Connection_Complete` event if the command is used to set up a new synchronous connection.

When used to modify an existing synchronous logical transport, only the `Packet_Type`, `Retransmission_Effort` and `Max_Latency` parameters may be modified.

Note: The Link Manager may choose any combination of packet types, timing, and retransmission window sizes that satisfy the parameters given. This may be achieved by using more frequent transmissions of smaller packets. The link manager may choose to set up either a SCO or an eSCO connection, if the parameters allow, using the corresponding LMP sequences.

Note: To modify a SCO connection, use the `HCI_Change_Connection_Packet_Type` command.

If the lower layers cannot achieve the exact transmit and receive bandwidth requested subject to the other parameters, or cannot achieve the transcoding or resampling implied by the parameters, then the link creation or link



modification shall be rejected. A synchronous connection may only be created when an ACL connection already exists.

The data at the audio data transport interface shall be treated as a stream of bits. The bits in each unit of data delivered by the transport shall be taken LSB first, and the units shall be taken in the order of delivery. The samples, encoded samples, frames, or other entity to be transcoded for transmission, or that has been transcoded after reception, shall be taken in the order of transmission or reception, with each entity taken LSB first.

For example, if the audio data transport uses 16 bit units and the Input or Output coding format is A-law, each unit represents two samples with the first in the 8 least significant bits and the second in the 8 most significant bits. Similarly, if the audio data transport uses 8 bit units and the Input or Output coding format is linear PCM with a size of 16 bits, the 8 least significant bits of each sample are transmitted first.

Command parameters:

Connection_Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xXXXX	Connection_Handle Range: 0x0000 to 0x0EFF

Transmit_Bandwidth: *Size: 4 octets*

Value	Parameter Description
0x00000000 to 0xFFFFFFFF	Transmit bandwidth in octets per second.
0xFFFFFFFF	Don't care

Receive_Bandwidth: *Size: 4 octets*

Value	Parameter Description
0x00000000 to 0xFFFFFFFF	Receive bandwidth in octets per second.
0xFFFFFFFF	Don't care

Transmit_Coding_Format: *Size: 5 octets*

Value	Parameter Description
Octet 0	See Assigned Numbers for Coding_Format
Octets 1 to 4	Octet 1 to 2: Company ID, see Assigned Numbers for Company Identifier Octet 3 to 4: Vendor specific codec ID. Shall be ignored if octet 0 of Transmit_Coding_Format is not 0xFF.

**Receive_Coding_Format:****Size: 5 octets**

Value	Parameter Description
Octet 0	See Assigned Numbers for Coding_Format
Octets 1 to 4	Octet 1 to 2: Company ID, see Assigned Numbers for Company Identifier. Octet 3 to 4: Vendor specific codec ID. Shall be ignored if octet 0 of Receive_Coding_Format is not 0xFF.

Transmit_Codec_Frame_Size:**Size: 2 octets**

Value	Parameter Description
0xFFFF	Range: 0x0001 to 0xFFFF, the actual size of the over-the-air encoded frame in octets.

Receive_Codec_Frame_Size:**Size: 2 octets**

Value	Parameter Description
0xFFFF	Range: 0x0001 to 0xFFFF, the actual size of the over-the-air encoded frame in octets.

Input_Bandwidth:**Size: 4 octets**

Value	Parameter Description
0xFFFFFFFF	Host to Controller nominal data rate in octets per second.

Output_Bandwidth:**Size: 4 octets**

Value	Parameter Description
0xFFFFFFFF	Controller to Host nominal data rate in octets per second.

Input_Coding_Format:**Size: 5 octets**

Value	Parameter Description
Octet 0	See Assigned Numbers for Coding_Format
Octets 1 to 4	Octet 1 to 2: Company ID, see Assigned Numbers for Company Identifier. Octet 3 to 4: Vendor specific codec ID. Shall be ignored if octet 0 of Input_Coding_Format is not 0xFF.

**Output_Coding_Format:****Size: 5 octets**

Value	Parameter Description
Octet 0	See Assigned Numbers for Coding_Format
Octets 1 to 4	Octet 1 to 2: Company ID, see Assigned Numbers for Company Identifier. Octet 3 to 4: Vendor specific codec ID. Shall be ignored if octet 0 of Output_Coding_Format is not 0xFF.

Input_Coded_Data_Size:**Size: 2 octets**

Value	Parameter Description
0xFFFF	Size, in bits, of the sample or framed data

Output_Coded_Data_Size:**Size: 2 octets**

Value	Parameter Description
0xFFFF	Size, in bits, of the sample or framed data

Input_PCM_Data_Format:**Size: 1 octet**

Value	Parameter Description
0xFF	See Assigned Numbers for PCM_Data_Format

Output_PCM_Data_Format:**Size: 1 octet**

Value	Parameter Description
0xFF	See Assigned Numbers for PCM_Data_Format

Input_PCM_Sample_Payload_MSB_Position:**Size: 1 octet**

Value	Parameter Description
0xFF	The number of bit positions within an audio sample that the MSB of the sample is away from starting at the MSB of the data.

Output_PCM_Sample_Payload_MSB_Position:**Size: 1 octet**

Value	Parameter Description
0xFF	The number of bit positions within an audio sample that the MSB of the sample is away from starting at the MSB of the data.

*Input_Data_Path:**Size: 1 octet*

Value	Parameter Description
0x00	HCI
0x01 to 0xFE	Logical_Channel_Number. The meaning of the logical channels will be vendor specific.
0xFF	Audio test mode

*Output_Data_Path:**Size: 1 octet*

Value	Parameter Description
0x00	HCI
0x01 to 0xFE	Logical_Channel_Number. The meaning of the logical channels will be vendor specific.
0xFF	Audio test mode

*Input_Transport_Unit_Size:**Size: 1 octet*

Value	Parameter Description
1 to 255	The number of bits in each unit of data received from the Host over the audio data transport.
0	Not applicable (implied by the choice of audio data transport)

*Output_Transport_Unit_Size:**Size: 1 octet*

Value	Parameter Description
1 to 255	The number of bits in each unit of data sent to the Host over the audio data transport.
0	Not applicable (implied by the choice of audio data transport)

*Max_Latency:**Size: 2 octets*

Value	Parameter Description
0x0000 to 0x0003	Reserved for future use
0x0004 to 0xFFFFE	The value in milliseconds representing the upper limit of the sum of the synchronous interval, and the size of the eSCO window, where the eSCO window is the reserved slots plus the retransmission window. (See [Vol 2] Part B, Figure 8.9)
0xFFFF	Don't care.

*Packet_Type:**Size: 2 octets*

Bit Number	Parameter Description
0	HV1 may be used
1	HV2 may be used
2	HV3 may be used
3	EV3 may be used
4	EV4 may be used
5	EV5 may be used
6	2-EV3 shall not be used
7	3-EV3 shall not be used
8	2-EV5 shall not be used
9	3-EV5 shall not be used
All other bits	Reserved for future use

Note: 0x003F means all packet types may be used.

*Retransmission_Effort:**Size: 1 octet*

Value	Parameter Description
0x00	No retransmission (SCO or eSCO connection allowed)
0x01	At least one retransmission, optimize for power consumption (eSCO connection required)
0x02	At least one retransmission, optimize for link quality (eSCO connection required)
0xFF	Don't care (SCO or eSCO connection allowed)
All other values	Reserved for future use

Return parameters:

None.

Event(s) generated (unless masked away):

When the BR/EDR Controller receives the HCI_Enhanced_Setup_-Synchronous_Connection command, it shall send the HCI_Command_Status event to the Host. In addition, when the LM determines that the connection is established, the local BR/EDR Controller shall send an HCI_Synchronous_-Connection_Complete event to the local Host, and the remote Controller will send an HCI_Synchronous_Connection_Complete event or an HCI_Connection_Complete event to the remote Host. The



HCI_Synchronous_Connection_Complete event contains the Connection_Handle if this command is successful.

This command cannot be used to change the parameters of a SCO link.



7.1.46 Enhanced Accept Synchronous Connection Request command

Command	OCF	Command Parameters	Return Parameters
HCI_Enhanced_Accept_Synchronous_Connection_Request	0x003E	BD_ADDR, Transmit_Bandwidth, Receive_Bandwidth, Transmit_Coding_Format, Receive_Coding_Format, Transmit_Codec_Frame_Size, Receive_Codec_Frame_Size, Input_Bandwidth, Output_Bandwidth, Input_Coding_Format, Output_Coding_Format, Input_Coded_Data_Size, Output_Coded_Data_Size, Input_PCM_Data_Format, Output_PCM_Data_Format, Input_PCM_Sample_Payload_MSB_Position, Output_PCM_Sample_Payload_MSB_Position, Input_Data_Path, Output_Data_Path, Input_Transport_Unit_Size, Output_Transport_Unit_Size, Max_Latency, Packet_Type, Retransmission_Effort	

Description:

The HCI_Enhanced_Accept_Synchronous_Connection_Request command is used to accept an incoming request for a synchronous connection and to present the local Link Manager with the acceptable parameter values for the synchronous connection. This command shall only be issued after an HCI_Connection_Request event, with link type SCO or eSCO, has occurred. An HCI_Connection_Request event contains the BD_ADDR of the device requesting the connection. The command to accept a connection must be received by the Controller before the timer Connection_Accept_Timeout expires on the local device.

The parameter set of the HCI_Enhanced_Accept_Synchronous_Connection_Request command is the same as for the HCI_Enhanced_



Setup_Synchronous_Connection command except for the Connection_Handle in the HCI_Enhanced_Setup_Synchronous_Connection command, which is replaced by the BD_ADDR in the HCI_Enhanced_Accept_Synchronous_Connection_Request command. See [Section 7.1.45](#) for the descriptions of these parameters.

The Host shall include in the Packet_Type parameter at least one packet type for the transport (SCO or eSCO) specified in the incoming request. The Controller shall ignore any packet types in the Packet_Type parameter for the other transport.

If the Link Type of the incoming request is SCO, then the Retransmission Effort parameter shall be ignored.

Note: See [Section 7.3.3](#) for the behavior when the HCI_Connection_Request event is masked or the connection is auto accepted.

If the ACL link has encryption enabled using AES-CCM then the Host shall not accept a request where the link type is SCO.

Command parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0XXXXXXXXXX X	BD_ADDR of the device requesting the connection

Transmit_Bandwidth:

Size: 4 octets

Value	Parameter Description
0x00000000 to 0xFFFFFFFFE	Transmit bandwidth in octets per second.
0xFFFFFFFF	Don't care

Receive_Bandwidth:

Size: 4 octets

Value	Parameter Description
0x00000000 to 0xFFFFFFFFE	Receive bandwidth in octets per second.
0xFFFFFFFF	Don't care

**Transmit_Coding_Format:****Size: 5 octets**

Value	Parameter Description
Octet 0	See Assigned Numbers for Coding_Format
Octets 1 to 4	Octet 1 to 2: Company ID, see Assigned Numbers for Company Identifier. Octet 3 to 4: Vendor specific codec ID. Shall be ignored if octet 0 of Transmit_Coding_Format is not 0xFF.

Receive_Coding_Format:**Size: 5 octets**

Value	Parameter Description
Octet 0	See Assigned Numbers for Coding_Format
Octets 1 to 4	Octet 1 to 2: Company ID, see Assigned Numbers for Company Identifier. Octet 3 to 4: Vendor specific codec ID. Shall be ignored if octet 0 of Receive_Coding_Format is not 0xFF.

Transmit_Codec_Frame_Size:**Size: 2 octets**

Value	Parameter Description
0xFFFF	Range: 0x0001 to 0xFFFF, the actual size of the over-the-air encoded frame in octets.

Receive_Codec_Frame_Size:**Size: 2 octets**

Value	Parameter Description
0xFFFF	Range: 0x0001 to 0xFFFF, the actual size of the over-the-air encoded frame in octets.

Input_Bandwidth:**Size: 4 octets**

Value	Parameter Description
0xFFFFFFFF	Host to Controller nominal data rate in octets per second.

Output_Bandwidth:**Size: 4 octets**

Value	Parameter Description
0xFFFFFFFF	Controller to Host nominal data rate in octets per second.

*Input_Coding_Format:**Size: 5 octets*

Value	Parameter Description
Octet 0	See Assigned Numbers for Coding_Format
Octets 1 to 4	Octet 1 to 2: Company ID, see Assigned Numbers for Company Identifier. Octet 3 to 4: Vendor specific codec ID. Shall be ignored if octet 0 of Input_Coding_Format is not 0xFF.

*Output_Coding_Format:**Size: 5 octets*

Value	Parameter Description
Octet 0	See Assigned Numbers for Coding_Format
Octets 1 to 4	Octet 1 to 2: Company ID, see Assigned Numbers for Company Identifier. Octet 3 to 4: Vendor specific codec ID. Shall be ignored if octet 0 of Output_Coding_Format is not 0xFF.

*Input_Coded_Data_Size:**Size: 2 octets*

Value	Parameter Description
0xFFFF	Size, in bits, of the sample or framed data

*Output_Coded_Data_Size:**Size: 2 octets*

Value	Parameter Description
0xFFFF	Size, in bits, of the sample or framed data

*Input_PCM_Data_Format:**Size: 1 octet*

Value	Parameter Description
0xFF	See Assigned Numbers for PCM_Data_Format

*Output_PCM_Data_Format:**Size: 1 octet*

Value	Parameter Description
0xFF	See Assigned Numbers for PCM_Data_Format

*Input_PCM_Sample_Payload_MSB_Position:**Size: 1 octet*

Value	Parameter Description
0xFF	The number of bit positions within an audio sample that the MSB of the sample is away from starting at the MSB of the data.

*Output_PCM_Sample_Payload_MSB_Position:**Size: 1 octet*

Value	Parameter Description
0xXX	The number of bit positions within an audio sample that the MSB of the sample is away from starting at the MSB of the data.

*Input_Data_Path:**Size: 1 octet*

Value	Parameter Description
0x00	HCI
0x01 to 0xFE	Logical_Channel_Number. The meaning of the logical channels will be vendor specific.
0xFF	Audio test mode

*Output_Data_Path:**Size: 1 octet*

Value	Parameter Description
0x00	HCI
0x01 to 0xFE	Logical_Channel_Number. The meaning of the logical channels will be vendor specific.
0xFF	Audio test mode

*Input_Transport_Unit_Size:**Size: 1 octet*

Value	Parameter Description
1 to 255	The number of bits in each unit of data received from the Host over the audio data transport.
0	Not applicable (implied by the choice of audio data transport)

*Output_Transport_Unit_Size:**Size: 1 octet*

Value	Parameter Description
1 to 255	The number of bits in each unit of data sent to the Host over the audio data transport.
0	Not applicable (implied by the choice of audio data transport)

*Max_Latency:**Size: 2 octets*

Value	Parameter Description
0x0000 to 0x0003	Reserved for future use
0x0004 to 0xFFFFE	The value in milliseconds representing the upper limit of the sum of the synchronous interval, and the size of the eSCO window, where the eSCO window is the reserved slots plus the retransmission window. (See [Vol 2] Part B, Figure 8.9)
0xFFFF	Don't care.

*Packet_Type:**Size: 2 octets*

Bit Number	Parameter Description
0	HV1 may be used
1	HV2 may be used
2	HV3 may be used
3	EV3 may be used
4	EV4 may be used
5	EV5 may be used
6	2-EV3 shall not be used
7	3-EV3 shall not be used
8	2-EV5 shall not be used
9	3-EV5 shall not be used
All other bits	Reserved for future use

Note: 0x003F means all packet types may be used.

*Retransmission_Effort:**Size: 1 octet*

Value	Parameter Description
0x00	No retransmission (SCO or eSCO connection allowed)
0x01	At least one retransmission, optimize for power consumption (eSCO connection required)
0x02	At least one retransmission, optimize for link quality (eSCO connection required)
0xFF	Don't care (SCO or eSCO connection allowed)
All other values	Reserved for future use

Return parameters:

None.

Event(s) generated (unless masked away):

The HCI_Enhanced_Accept_Synchronous_Request command requests the local BR/EDR Controller to start setting up the connection. When this action commences, the HCI_Command_Status event shall be sent by the BR/EDR Controller. When the link setup is complete, the BR/EDR Controller shall send an HCI_Synchronous_Connection_Complete event to its Host, and the remote BR/EDR Controller will send an HCI_Synchronous_Connection_Complete event to its Host. The HCI_Synchronous_Connection_Complete will contain the Connection_Handle and the link parameters if the setup is successful.



7.1.47 Truncated Page command

Command	OCF	Command Parameters	Return Parameters
HCI_Truncated_Page	0x003F	BD_ADDR, Page_Scan_Repetition_Mode, Clock_Offset	

Description:

The HCI_Truncated_Page command is used to page the BR/EDR Controller with the specified BD_ADDR and then abort the paging sequence after an ID response has been received. See [\[Vol 2\] Part B, Section 8.3.3](#) for additional information.

The Page_Scan_Repetition_Mode parameter specifies the Page Scan Repetition mode supported by the remote BR/EDR Controller with the BD_ADDR. This is the most recent version of the information that was acquired either during the inquiry process or from an HCI_Page_Scan_Repetition_Mode_Change event (see [Section 7.7.31](#)).

The Clock_Offset parameter is the difference between the local BR/EDR Controller's own clock and the clock of the remote BR/EDR Controller with BD_ADDR. Only bits 2 to 16 of the difference are used, and they are mapped to this parameter as bits 0 to 14 respectively. A Clock_Offset_Valid_Flag, located in bit 15 of the Clock_Offset parameter, indicates if the Clock Offset is valid or not.

Command parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0xFFFFFFFFXXXX	BD_ADDR of the Device to page

Page_Scan_Repetition_Mode:

Size: 1 octet

Value	Parameter Description
0x00	R0
0x01	R1
0x02	R2
All other values	Reserved for future use.

**Clock_Offset:****Size: 2 octets**

Bit Number	Parameter Description
0 to 14	Bits 2 to 16 of CLKNPeripheral - CLK
15	Clock Offset is valid

Return parameters:

None.

Event(s) generated (unless masked away):

When the BR/EDR Controller receives the HCI_Truncated_Page command the BR/EDR Controller shall send the HCI_Command_Status event to the Host. In addition, when the Truncated Page procedure has completed, the BR/EDR Controller shall send an HCI_Truncated_Page_Complete event to the Host.



7.1.48 Truncated Page Cancel command

Command	OCF	Command Parameters	Return Parameters
HCI_Truncated_Page_Cancel	0x0040	BD_ADDR	Status, BD_ADDR

Description:

The HCI_Truncated_Page_Cancel command is used to request cancellation of an ongoing Truncated_Page process previously started by an HCI_Truncated_Page command.

Command parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0xFFFFFFFFXXXX	BD_ADDR of the device to which the HCI_Truncated_Page command was previously issued and that is the subject of the cancellation request.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Truncated_Page_Cancel command succeeded.
0x01 to 0xFF	HCI_Truncated_Page_Cancel command failed. See [Vol 1] Part F, Controller Error Codes , for error codes and descriptions

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0xFFFFFFFFXXXX	BD_ADDR of the device to which the HCI_Truncated_Page command was previously issued before and that is the subject of the cancellation request.

Event(s) generated (unless masked away):

When the HCI_Truncated_Page_Cancel command has completed, an HCI_Command_Complete event shall be generated.

If the truncated page procedure has already completed, but the BR/EDR Controller has not yet sent the HCI_Truncated_Page_Complete event, then the local device shall return an HCI_Command_Complete event with status "Success".



If the HCI_Truncated_Page_Cancel command is sent to the BR/EDR Controller without a pending HCI_Truncated_Page command to the same device, the BR/EDR Controller shall return an HCI_Command_Complete event with the error code *Unknown Connection Identifier* (0x02).

Note: From the BR/EDR Controller perspective this is identical to the situation where the HCI_Truncated_Page command has already completed and the HCI_Truncated_Page_Complete event already sent.



7.1.49 Set Connectionless Peripheral Broadcast command

Command	OCF	Command Parameters	Return Parameters
HCI_Set_Connectionless_Peripheral_Broadcast	0x0041	Enable, LT_ADDR, LPO_Allowed, Packet_Type, Interval_Min, Interval_Max, Supervision_Timeout	Status, LT_ADDR, Interval

Description:

The HCI_Set_Connectionless_Peripheral_Broadcast command controls the Connectionless Peripheral Broadcast functionality in the BR/EDR Controller. Connectionless Peripheral Broadcast mode may be enabled or disabled by the Enable parameter. If Enable is set to 0x00 and the Synchronization Train substate is active, then the Controller shall also exit the Synchronization Train substate. If Enable is set to 0x01, the remaining parameters shall be ignored.

The LT_ADDR indicated in the Set_Connectionless_Peripheral_Broadcast shall be pre-allocated using the HCI_Set_Reserved_LT_ADDR command. If the LT_ADDR has not been reserved, the *Unknown Connection Identifier* (0x02) error code shall be returned. If the Controller is unable to reserve sufficient bandwidth for the requested activity, the *Connection Rejected Due to Limited Resources* (0x0D) error code shall be returned.

The LPO_Allowed parameter informs the BR/EDR Controller whether it is allowed to sleep.

The Packet_Type parameter specifies which packet types are allowed. The Host shall either enable BR packet types only, or shall enable EDR and DM1 packet types only.

The Interval_Min and Interval_Max parameters specify the range from which the BR/EDR Controller shall select the Connectionless Peripheral Broadcast Interval. The selected Interval is returned.

Command parameters:

Enable:

Size: 1 octet

Value	Parameter Description
0x00	Disabled
0x01	Enabled
All other values	Reserved for future use

**LT_ADDR:****Size: 1 octet**

Value	Parameter Description
0x01 to 0x07	LT_ADDR used for Connectionless Peripheral Broadcast
All other values	Reserved for future use

LPO_Allowed:**Size: 1 octet**

Value	Parameter Description
0x00	BR/EDR Controller shall not sleep (that is, clock accuracy shall be equal to or better than ± 20 ppm)
0x01	BR/EDR Controller may sleep (that is, clock accuracy shall be equal to or better than ± 250 ppm)
All other values	Reserved for future use

Packet_Type:**Size: 2 octets**

Bit Number	Parameter Description
1	2-DH1 shall not be used
2	3-DH1 shall not be used
3	DM1 may be used
4	DH1 may be used
8	2-DH3 shall not be used
9	3-DH3 shall not be used
10	DM3 may be used
11	DH3 may be used
12	2-DH5 shall not be used
13	3-DH5 shall not be used
14	DM5 may be used
15	DH5 may be used
All other bits	Reserved for future use.

*Interval_Min:**Size: 2 octets*

Value	Parameter Description
0xXXXX	Minimum interval between Connectionless Peripheral Broadcast packets in slots. Range: 0x0002 to 0xFFFE; only even values are valid

*Interval_Max:**Size: 2 octets*

Value	Parameter Description
0xXXXX	Maximum interval between Connectionless Peripheral Broadcast packets in slots. Range: 0x0002 to 0xFFFE; only even values are valid

*Supervision_Timeout:**Size: 2 octets*

Value	Parameter Description
0xXXXX	Duration in slots after which the BR/EDR Controller reports an HCI_Connectionless_Peripheral_Broadcast_Timeout event if it is unable to transmit a Connectionless Peripheral Broadcast packet. Range: 0x0002 to 0xFFFE; only even values are valid

Return parameters:*Status:**Size: 1 octet*

Value	Parameter Description
0x00	HCI_Set_Connectionless_Peripheral_Broadcast command succeeded.
0x01 to 0xFF	HCI_Set_Connectionless_Peripheral_Broadcast command failed. See [Vol 1] Part F, Controller Error Codes , for error codes and descriptions.

*LT_ADDR:**Size: 1 octet*

Value	Parameter Description
0x01 to 0x07	LT_ADDR used for Connectionless Peripheral Broadcast
All other values	Reserved for future use

*Interval:**Size: 2 octets*

Value	Parameter Description
0xXXXX	Actual interval between Connectionless Peripheral Broadcast packets in slots. Range: 0x0002 to 0xFFFE; only even values are valid

**Event(s) generated (unless masked away):**

When the HCI_Set_Connectionless_Peripheral_Broadcast command has completed, an HCI_Command_Complete event shall be generated.

If the BR/EDR Controller is unable to transmit a Connectionless Peripheral Broadcast packet for *Supervision_Timeout* slots, it shall generate an HCI_Connectionless_Peripheral_Broadcast_Timeout event.



7.1.50 Set Connectionless Peripheral Broadcast Receive command

Command	OCF	Command Parameters	Return Parameters
HCI_Set_Connectionless_Peripheral_Broadcast_Receive	0x0042	Enable, BD_ADDR, LT_ADDR, Interval, Clock_Offset, Next_Connectionless_Peripheral_Broadcast_Clock, Supervision_Timeout, Remote_Timing_Accuracy, Skip, Packet_Type, AFH_Channel_Map	Status, BD_ADDR, LT_ADDR

Description:

The HCI_Set_Connectionless_Peripheral_Broadcast_Receive command controls the reception of Connectionless Peripheral Broadcast packets in the BR/EDR Controller of a Connectionless Peripheral Broadcast Receiver. If the Enable parameter is set to Disabled, the BR/EDR Controller does not attempt to receive Connectionless Peripheral Broadcast packets and the remaining parameters shall be ignored. If the Enable parameter is set to Enabled, the BR/EDR Controller starts attempting to receive Connectionless Peripheral Broadcast packets on the specified LT_ADDR.

The Interval parameter specifies the interval of the Connectionless Peripheral Broadcast to be used by the BR/EDR Controller.

The Skip parameter specifies the number of consecutive Connectionless Peripheral Broadcast instants which the receiver may skip after successfully receiving a Connectionless Peripheral Broadcast packet.

The Packet_Type parameter specifies which packet types are allowed. The Host shall either enable BR packet types only or shall enable EDR and DM1 packet types only.

The AFH_Channel_Map parameter is the AFH channel map used by the Transmitter for the PBD logical link, and is obtained by the Receiver's Host from the HCI_Synchronization_Train_Received event.

**Command parameters:***Enable:**Size: 1 octet*

Value	Parameter Description
0x00	Disabled
0x01	Enabled
All other values	Reserved for future use

*BD_ADDR:**Size: 6 octets*

Value	Parameter Description
0XXXXXXXXXXXXX	BD_ADDR of the Connectionless Peripheral Broadcast transmitter

*LT_ADDR:**Size: 1 octet*

Value	Parameter Description
0x01 to 0x07	LT_ADDR to use for receiving Connectionless Peripheral Broadcast messages
All other values	Reserved for future use

*Interval:**Size: 2 octets*

Value	Parameter Description
0XXXXX	Interval between Connectionless Peripheral Broadcast packets instants in slots. Range: 0x0002 to 0xFFFE; only even values are valid

*Clock_Offset:**Size: 4 octets (28 bits meaningful)*

Value	Parameter Description
0XXXXXXXX	$(\text{CLKN}_{\text{receiver}} - \text{CLKN}_{\text{transmitter}}) \bmod 2^{28}$

*Next_Connectionless_Peripheral_-
Broadcast_Clock:**Size: 4 octets (28 bits meaningful)*

Value	Parameter Description
0XXXXXXXX	CLK for next Connectionless Peripheral Broadcast instant

*Supervision_Timeout:**Size: 2 octets*

Value	Parameter Description
0XXXXX	Duration in slots to continue listening for Connectionless Peripheral Broadcast packets after the last successfully received Connectionless Peripheral Broadcast packet. Range: 0x0002 to 0xFFFE; only even values are valid

**Remote_Timing_Accuracy:****Size: 1 octet**

Value	Parameter Description
0xXX	Timing accuracy of the Transmitter in ppm. Typical values are 20 ppm and 250 ppm.

Skip:**Size: 1 octet**

Value	Parameter Description
0xXX	Number of Connectionless Peripheral Broadcast instants to skip after successfully receiving a Broadcast packet.

Packet_Type:**Size: 2 octets**

Bit Number	Parameter Description
1	2-DH1 shall not be used
2	3-DH1 shall not be used
3	DM1 may be used
4	DH1 may be used
8	2-DH3 shall not be used
9	3-DH3 shall not be used
10	DM3 may be used
11	DH3 may be used
12	2-DH5 shall not be used
13	3-DH5 shall not be used
14	DM5 may be used
15	DH5 may be used
All other bits	Reserved for future use.

AFH_Channel_Map:**Size: 10 octets (79 bits meaningful)**

Value	Parameter Description
0XXXXXXXXX XXXXXXXXXX XX	<p>This parameter contains 80 1-bit fields.</p> <p>The n^{th} such field (in the range 0 to 78) contains the value for channel n:</p> <p>0: channel n is unused 1: channel n is used</p> <p>The most significant bit (bit 79) is reserved for future use</p> <p>At least N_{min} channels shall be marked as used (see [Vol 2] Part B, Section 2.3.1)</p>

**Return parameters:***Status:**Size: 1 octet*

Value	Parameter Description
0x00	HCI_Set_Connectionless_Peripheral_Broadcast_Receive command succeeded.
0x01 to 0xFF	HCI_Set_Connectionless_Peripheral_Broadcast_Receive command failed. See [Vol 1] Part F, Controller Error Codes , for error codes and descriptions.

*BD_ADDR:**Size: 6 octets*

Value	Parameter Description
0XXXXXXXXXXXXX	BD_ADDR of the Connectionless Peripheral Broadcast transmitter

*LT_ADDR:**Size: 1 octet*

Value	Parameter Description
0x01 to 0x07	LT_ADDR used for receiving Connectionless Peripheral Broadcast messages
All other values	Reserved for future use

Event(s) generated (unless masked away):

When the HCI_Set_Connectionless_Peripheral_Broadcast_Receive command has been received, an HCI_Command_Complete event shall be generated. Completion of the HCI_Set_Connectionless_Peripheral_Broadcast_Receive command does not require reception of a Connectionless Peripheral Broadcast packet. If the BR/EDR Controller does not receive a Connectionless Peripheral Broadcast packet for *CPB_supervisionTO* slots, it shall generate an HCI_Connectionless_Peripheral_Broadcast_Timeout event.



7.1.51 Start Synchronization Train command

Command	OCF	Command Parameters	Return Parameters
HCI_Start_Synchronization_Train	0x0043		

Description:

The HCI_Start_Synchronization_Train command controls the Synchronization Train functionality in the BR/EDR Controller. Connectionless Peripheral Broadcast mode shall be enabled on the BR/EDR Controller before this command may be used. If Connectionless Peripheral Broadcast mode is not enabled, the *Command Disallowed* (0x0C) error code shall be returned. After receiving this command and returning an HCI_Command_Status event, the Baseband starts attempting to send synchronization train packets containing information related to the enabled Connectionless Peripheral Broadcast packet timing.

Note: The AFH_Channel_Map used in the synchronization train packets is configured by the HCI_Set_AFH_Host_Channel_Classification command and the local channel classification in the BR/EDR Controller.

The synchronization train packets will be sent using the parameters specified by the latest HCI_Write_Synchronization_Train_Parameters command. The Synchronization Train will continue until *synchronization_trainTO* slots (as specified in the last HCI_Write_Synchronization_Train_Parameters command) have passed or until the Host disables the Connectionless Peripheral Broadcast logical transport.

Command parameters:

None.

Return parameters:

None

Event(s) generated (unless masked away):

When the BR/EDR Controller receives the HCI_Start_Synchronization_Train command, it shall send an HCI_Command_Status event to the Host.



7.1.52 Receive Synchronization Train command

Command	OCF	Command Parameters	Return Parameters
HCI_Receive_ Synchronization_Train	0x0044	BD_ADDR, Sync_Scan_Timeout, Sync_Scan_Window, Sync_Scan_Interval	

Description:

The HCI_Receive_Synchronization_Train command requests synchronization with the specified Connectionless Peripheral Broadcast Transmitter. The Sync_Scan_Window parameter specifies the duration of each scan and the Sync_Scan_Interval parameter specifies the interval between the start of consecutive scan windows. An HCI_Synchronization_Train_Received event shall be sent if a synchronization train packet is received with a non-zero Connectionless Peripheral Broadcast LT_ADDR or if the BR/EDR Controller fails to receive a synchronization train packet for *synchronization_scanTO* slots.

Command parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0XXXXXXXXXXXXX	BD_ADDR of the Connectionless Peripheral Broadcast transmitter

Sync_Scan_Timeout:

Size: 2 octets

Value	Parameter Description
0XXXXX	Duration in slots to search for the synchronization train Shall be greater than or equal to Sync_Scan_Window; only even values are valid

Sync_Scan_Window:

Size: 2 octets

Value	Parameter Description
0XXXXX	Duration in slots to listen for a synchronization train packet on a single frequency Range: 0x0022 to 0xFFFC; only even values are valid

**Sync_Scan_Interval:****Size: 2 octets**

Value	Parameter Description
0xFFFF	Duration in slots between the start of consecutive scan windows Shall be greater than or equal to Sync_Scan_Window+0x0002; only even values are valid

Return parameters:

None.

Event(s) generated (unless masked away):

When the BR/EDR Controller receives the HCI_Receive_Synchronization_Train command, it shall send an HCI_Command_Status event to the Host. In addition, when the BR/EDR Controller receives, or fails to receive within the duration specified by *Sync_Scan_Timeout*, the synchronization train from the Connectionless Peripheral Broadcast transmitter, it shall send an HCI_Synchronization_Train_Received event to the Host.



7.1.53 Remote OOB Extended Data Request Reply command

Command	OCF	Command Parameters	Return Parameters
HCI_Remote_OOB_Extended_Data_Request_Reply	0x0045	BD_ADDR, C_192, R_192, C_256, R_256	Status, BD_ADDR

Description:

The HCI_Remote_OOB_Extended_Data_Request_Reply command is used to reply to an HCI_Remote_OOB_Data_Request event with the C and R values derived with the P-192 public key and the C and R values associated with the P-256 public key received via an OOB transfer from a remote device identified by BD_ADDR. If the C_192 and R_192 values are not present in the received OOB data from the remote device, the Host shall set C_192 and R_192 to zeros. If the C_256 and R_256 values are not present in the received OOB data from the remote device, the Host shall set C_256 and R_256 to zeros.

Command parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0XXXXXXXXXXXXX X	BD_ADDR of remote device from which the C and R values were received

C_192:

Size: 16 octets

Value	Parameter Description
0XXXXXXXXXXXXX XXXXXXXXXXXXX XXXXXXXXXXXXX	Secure Simple Pairing Hash C derived from the P-192 public key.

R_192:

Size: 16 octets

Value	Parameter Description
0XXXXXXXXXXXXX XXXXXXXXXXXXX XXXXXXXXXXXXX	Secure Simple Pairing Randomizer associated with the P-192 public key.

**C_256:****Size: 16 octets**

Value	Parameter Description
0XXXXXXXXXXXXX XXXXXXXXXXXXX XXXXXXXXXXXXX	Secure Simple Pairing Hash C derived from the P-256 public key.

R_256:**Size: 16 octets**

Value	Parameter Description
0XXXXXXXXXXXXX XXXXXXXXXXXXX XXXXXXXXXXXXX	Secure Simple Pairing Randomizer associated with the P-256 public key.

Return parameters:**Status:****Size: 1 octet**

Value	Parameter Description
0x00	HCI_Remote_OOB_Extended_Data_Request_Reply command succeeded.
0x01 to 0xFF	HCI_Remote_OOB_Extended_Data_Request_Reply command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

BD_ADDR:**Size: 6 octets**

Value	Parameter Description
0XXXXXXXXXXXXX X	BD_ADDR of remote device from which the C and R values were received

Event(s) generated (unless masked away):

When the HCI_Remote_OOB_Extended_Data_Request_Reply command has completed, an HCI_Command_Complete event shall be generated.



7.2 LINK POLICY COMMANDS

The Link Policy commands provide methods for the Host to affect how the Link Manager manages the piconet. When Link Policy commands are used, the LM still controls how Bluetooth piconets and scatternets are established and maintained, depending on adjustable policy parameters. These policy commands modify the Link Manager behavior that can result in changes to the Link Layer connections with Bluetooth remote devices.

Note: Only one ACL connection can exist between two BR/EDR Controllers, and therefore there can only be one ACL HCI Connection_Handle for each physical Link Layer Connection. The BR/EDR Controller provides policy adjustment mechanisms to provide support for a number of different policies. This capability allows one Bluetooth module to be used to support many different usage models, and the same Bluetooth module can be incorporated in many different types of BR/EDR Controllers.

For the Link Policy commands, the OGF is defined as 0x02.

7.2.1 Hold Mode command

Command	OCF	Command Parameters	Return Parameters
HCI_Hold_Mode	0x0001	Connection_Handle, Hold_Mode_Max_Interval, Hold_Mode_Min_Interval	

Description:

The HCI_Hold_Mode command is used to alter the behavior of the Link Manager, and have it place the ACL Baseband connection associated by the specified Connection_Handle into the Hold mode. The Connection_Handle shall identify an ACL connection and not a SCO or eSCO connection. The Hold_Mode_Max_Interval and Hold_Mode_Min_Interval command parameters specify the length of time the Host wants to put the connection into the Hold mode. The local and remote devices will negotiate the length in the Hold mode. The Hold_Mode_Max_Interval parameter is used to specify the maximum length of the Hold interval for which the Host may actually enter into the Hold mode after negotiation with the remote device. The Hold interval defines the amount of time between when the Hold mode begins and when the Hold mode is completed. The Hold_Mode_Min_Interval parameter is used to specify the minimum length of the Hold interval for which the Host may actually enter into the Hold mode after the negotiation with the remote device. Therefore the Hold_Mode_Min_Interval shall not be greater than the Hold_Mode_Max_Interval. The BR/EDR Controller will return the actual Hold interval in the Interval parameter of the HCI_Mode_Change event, if the command is successful. This command enables the Host to support a low-power policy for itself or sev-



eral other BR/EDR Controllers, and allows the devices to enter Inquiry Scan, Page Scan, and a number of other possible actions.

Note: If the Host sends data to the BR/EDR Controller with a Connection_Handle corresponding to a connection in Hold mode, the BR/EDR Controller will keep the data in its buffers until either the data can be transmitted (the Hold mode has ended) or a flush, a flush timeout or a disconnection occurs. This is valid even if the Host has not yet been notified of the Hold mode through an HCI_Mode_Change event when it sends the data.

Note: The above is not valid for an HCI ACL Data packet sent from the Host to the BR/EDR Controller on the Central side where the Connection_Handle is a Connection_Handle used for broadcast and the Broadcast_Flag is set to Active Broadcast. The broadcast data will then never be received by Peripherals in Hold mode.

The Hold_Mode_Max_Interval shall be less than the Link Supervision Timeout configuration parameter.

Command parameters:

Connection_Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xXXXX	Connection_Handle Range: 0x0000 to 0x0EFF

Hold_Mode_Max_Interval:

Size: 2 octets

Value	Parameter Description
N = 0xXXXX	Maximum acceptable number of Baseband slots to wait in Hold mode. Time Length of the Hold = $N * 0.625$ ms (1 Baseband slot) Range: 0x0002 to 0xFFFE; only even values are valid. Time Range: 1.25 ms to 40.9 s Mandatory Range: 0x0014 to 0x8000

Hold_Mode_Min_Interval:

Size: 2 octets

Value	Parameter Description
N = 0xXXXX	Minimum acceptable number of Baseband slots to wait in Hold mode. Time Length of the Hold = $N * 0.625$ ms (1 Baseband slot) Range: 0x0002 to 0xFF00; only even values are valid Time Range: 1.25 ms to 40.9 s Mandatory Range: 0x0014 to 0x8000

**Return parameters:**

None.

Event(s) generated (unless masked away):

When the BR/EDR Controller receives the HCI_Hold_Mode command, the BR/EDR Controller shall send the HCI_Command_Status event to the Host. The HCI_Mode_Change event shall occur when the Hold mode has started and the HCI_Mode_Change event shall occur again when the Hold mode has completed for the specified Connection_Handle. The HCI_Mode_Change event signaling the end of the Hold mode is an estimation of the Hold mode ending if the event is for a remote BR/EDR Controller.



7.2.2 Sniff Mode command

Command	OCF	Command Parameters	Return Parameters
HCI_Sniff_Mode	0x0003	Connection_Handle, Sniff_Max_Interval, Sniff_Min_Interval, Sniff_Attempt, Sniff_Timeout	

Description:

The HCI_Sniff_Mode command is used to alter the behavior of the Link Manager and have it place the ACL Baseband connection associated with the specified Connection_Handle into Sniff mode. The Connection_Handle command parameter is used to identify which ACL link connection is to be placed in Sniff mode and shall identify an ACL connection, not a SCO or eSCO, connection. The Sniff_Max_Interval and Sniff_Min_Interval command parameters are used to specify the requested acceptable maximum and minimum periods in Sniff mode. The Sniff_Min_Interval shall not be greater than the Sniff_Max_Interval. The sniff interval defines the amount of time between each consecutive sniff period. The BR/EDR Controller will return the actual sniff interval in the Interval parameter of the HCI_Mode_Change event, if the command is successful. For a description of the meaning of the Sniff_Attempt and Sniff_Timeout parameters, see [\[Vol 2\] Part B, Section 8.7](#). Sniff_Attempt is there called $N_{\text{sniff attempt}}$ and Sniff_Timeout is called $N_{\text{sniff timeout}}$. This command enables the Host to support a low-power policy for itself or several other BR/EDR Controllers, and allows the devices to enter Inquiry Scan, Page Scan, and a number of other possible actions.

Note: If the Host sends data to the BR/EDR Controller with a Connection_Handle corresponding to a connection in Sniff mode, the BR/EDR Controller will keep the data in its buffers until either the data can be transmitted or a flush, a flush timeout or a disconnection occurs. This is valid even if the Host has not yet been notified of Sniff mode through an HCI_Mode_Change event when it sends the data.

Note: It is possible for the Central to transmit data to a Peripheral without exiting Sniff mode (see description in [\[Vol 2\] Part B, Section 8.7](#)).

Note: The above is not valid for an HCI ACL Data packet sent from the Host to the BR/EDR Controller on the Central side where the Connection_Handle is a Connection_Handle used for broadcast and the Broadcast_Flag is set to Active Broadcast. In that case, the broadcast data will only be received by a Peripheral in Sniff mode if that Peripheral happens to listen to the Central when the broadcast is made.

The Sniff_Max_Interval shall be less than the Link Supervision Timeout configuration parameter.

**Command parameters:***Connection_Handle:**Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xXXXX	Connection_Handle Range: 0x0000 to 0x0EFF

*Sniff_Max_Interval:**Size: 2 octets*

Value	Parameter Description
N = 0xXXXX	Range: 0x0002 to 0xFFFFE; only even values are valid Mandatory Range: 0x0006 to 0x0540 Time = N * 0.625 ms Time Range: 1.25 ms to 40.9 s

*Sniff_Min_Interval:**Size: 2 octets*

Value	Parameter Description
N = 0xXXXX	Range: 0x0002 to 0xFFFFE; only even values are valid Mandatory Range: 0x0006 to 0x0540 Time = N * 0.625 ms Time Range: 1.25 ms to 40.9 s

*Sniff_Attempt:**Size: 2 octets*

Value	Parameter Description
N = 0xXXXX	Number of Baseband receive slots for sniff attempt. Length = N * 1.25 ms Range: 0x0001 to 0x7FFF Time Range: 1.25 ms to 40.9 s Mandatory Range for Controller: 1 to $T_{\text{sniff}}/2$

*Sniff_Timeout:**Size: 2 octets*

Value	Parameter Description
N = 0xXXXX	Number of Baseband receive slots for sniff timeout. Length = N * 1.25 ms Range: 0x0000 to 0x7FFF Time Range: 0 ms to 40.9 s Mandatory Range for Controller: 0 to 0x0028

Return parameters:

None.

**Event(s) generated (unless masked away):**

When the BR/EDR Controller receives the HCI_Sniff_Mode command, the BR/EDR Controller shall send the HCI_Command_Status event to the Host. The HCI_Mode_Change event shall occur when Sniff mode has started for the specified Connection_Handle.



7.2.3 Exit Sniff Mode command

Command	OCF	Command Parameters	Return Parameters
HCI_Exit_Sniff_Mode	0x0004	Connection_Handle	

Description:

The HCI_Exit_Sniff_Mode command is used to end Sniff mode for a Connection_Handle, which is currently in Sniff mode. The Connection_Handle shall identify an ACL connection and not a SCO or eSCO connection. The Link Manager will determine and issue the appropriate LMP commands to remove Sniff mode for the associated Connection_Handle.

Command parameters:

Connection_Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xXXXX	Connection_Handle Range: 0x0000 to 0x0EFF

Return parameters:

None.

Event(s) generated (unless masked away):

When BR/EDR Controller receives the HCI_Exit_Sniff_Mode command, the BR/EDR Controller shall send the HCI_Command_Status event to the Host. The HCI_Mode_Change event shall occur when Sniff mode has ended for the specified Connection_Handle.



7.2.4 [This section is no longer used]

7.2.5 [This section is no longer used]



7.2.6 QoS Setup command

Command	OCF	Command Parameters	Return Parameters
HCI_QoS_Setup	0x0007	Connection_Handle, Unused, Service_Type, Token_Rate, Peak_Bandwidth, Latency, Delay_Variation	

Description:

The HCI_QoS_Setup command is used to specify Quality of Service parameters for a Connection_Handle. The Connection_Handle shall be a Connection_Handle for an ACL connection. These QoS parameter are the same parameters as L2CAP QoS. For more detail see [\[Vol 3\] Part A, Logical Link Control and Adaptation Protocol Specification](#). This allows the Link Manager to have all of the information about what the Host is requesting for each connection. The LM will determine if the QoS parameters can be met. BR/EDR Controllers that are both Peripherals and Centrals can use this command. When a device is a Peripheral, this command will trigger an LMP request to the Central to provide the Peripheral with the specified QoS as determined by the LM. When a device is a Central, this command is used to request a Peripheral to accept the specified QoS as determined by the LM of the Central. The Connection_Handle command parameter is used to identify for which connection the QoS request is requested.

The Unused parameter is reserved for future use.

Command parameters:

Connection_Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xXXXX	Connection_Handle Range: 0x0000 to 0x0EFF

Unused:

Size: 1 octet

Value	Parameter Description
0x00	This value shall be used by the Host.
All other values	Reserved for future use.

**Service_Type:****Size: 1 octet**

Value	Parameter Description
0x00	No Traffic.
0x01	Best Effort.
0x02	Guaranteed.
All other values	Reserved for future use.

Token_Rate:**Size: 4 octets**

Value	Parameter Description
0xFFFFFFFF	Token Rate in octets per second.

Peak_Bandwidth:**Size: 4 octets**

Value	Parameter Description
0xFFFFFFFF	Peak Bandwidth in octets per second.

Latency:**Size: 4 octets**

Value	Parameter Description
0xFFFFFFFF	Latency in microseconds.

Delay_Variation:**Size: 4 octets**

Value	Parameter Description
0xFFFFFFFF	Delay Variation in microseconds.

Return parameters:

None.

Event(s) generated (unless masked away):

When the BR/EDR Controller receives the HCI_QoS_Setup command, the BR/EDR Controller shall send the HCI_Command_Status event to the Host. When the Link Manager has completed the LMP messages to establish the requested QoS parameters, the BR/EDR Controller shall send an HCI_QoS_Setup_Complete event to the Host, and the event may also be generated on the remote side if there was LMP negotiation. The values of the parameters of the HCI_QoS_Setup_Complete event may, however, be different on the initiating and the remote side. The HCI_QoS_Setup_Complete event returned by the BR/EDR Controller on the local side contains the status of this command, and returned QoS parameters describing the supported QoS for the connection.

Note: If the Link Manager performs an LMP transaction that involves the flow parameter values on the remote side, the remote Controller can send an



HCI_Flow_Specification_Complete event or HCI_QoS_Setup_Complete event to the remote Host.



7.2.7 Role Discovery command

Command	OCF	Command Parameters	Return Parameters
HCI_Role_Discovery	0x0009	Connection_Handle	Status, Connection_Handle, Current_Role

Description:

The HCI_Role_Discovery command is used for a Host to determine which role the device is performing for a particular Connection_Handle. The Connection_Handle shall be a Connection_Handle for an ACL connection.

Command parameters:

Connection_Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xXXXX	Connection_Handle Range: 0x0000 to 0x0EFF

Return parameters:

Status: *Size: 1 octet*

Value	Parameter Description
0x00	HCI_Role_Discovery command succeeded
0x01 to 0xFF	HCI_Role_Discovery command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Connection_Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xXXXX	Connection_Handle Range: 0x0000 to 0x0EFF

Current_Role: *Size: 1 octet*

Value	Parameter Description
0x00	Current Role is Central for this Connection_Handle.
0x01	Current Role is Peripheral for this Connection_Handle.

Event(s) generated (unless masked away):

When the HCI_Role_Discovery command has completed, an HCI_Command_Complete event shall be generated.



7.2.8 Switch Role command

Command	OCF	Command Parameters	Return Parameters
HCI_Switch_Role	0x000B	BD_ADDR, Role	

Description:

The HCI_Switch_Role command is used to switch the current BR/EDR role the device is performing for a particular connection with another specified BR/EDR Controller. The BD_ADDR command parameter indicates for which connection the role switch is to be performed and shall specify a BR/EDR Controller for which a connection already exists. The Role parameter indicates the requested new role that the local device performs.

If there is an (e)SCO connection between the local device and the device identified by the BD_ADDR parameter, an attempt to perform a role switch shall be rejected by the local device.

If the connection between the local device and the device identified by the BD_ADDR parameter is placed in Sniff mode, an attempt to perform a role switch shall be rejected by the local device.

Command parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0XXXXXXXXXX XX	BD_ADDR for the connected device with which a role switch is to be performed.

Role:

Size: 1 octet

Value	Parameter Description
0x00	Change own Role to Central for this BD_ADDR.
0x01	Change own Role to Peripheral for this BD_ADDR.

Return parameters:

None.

Event(s) generated (unless masked away):

When the BR/EDR Controller receives the HCI_Switch_Role command, the BR/EDR Controller shall send the HCI_Command_Status event to the Host. When the role switch is performed, an HCI_Role_Change event shall occur to indicate that the roles have been changed, and will be communicated to both Hosts. If no change is required, only the Controller on the local device shall send the



event. If a Baseband role switch is attempted but fails, the local Controller shall send the event and the remote Controller may send it.



7.2.9 Read Link Policy Settings command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Link_Policy_Settings	0x000C	Connection_Handle	Status, Connection_Handle, Link_Policy_Settings

Description:

This command will read the Link Policy setting for the specified Connection_Handle. The Connection_Handle shall be a Connection_Handle for an ACL connection. [Section 6.18](#).

Command parameters:

Connection_Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

Return parameters:

Status: *Size: 1 octet*

Value	Parameter Description
0x00	HCI_Read_Link_Policy_Settings command succeeded.
0x01 to 0xFF	HCI_Read_Link_Policy_Settings command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Connection_Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

**Link_Policy_Settings:****Size: 2 octets**

Bit Number	Parameter Description
0	Enable Role Switch.
1	Enable Hold mode.
2	Enable Sniff mode.
All other bits	Reserved for future use.

Event(s) generated (unless masked away):

When the HCI_Read_Link_Policy_Settings command has completed, an HCI_Command_Complete event shall be generated.



7.2.10 Write Link Policy Settings command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Link_Policy_Settings	0x000D	Connection_Handle, Link_Policy_Settings	Status, Connection_Handle

Description:

This command writes the Link Policy setting for the specified Connection_Handle. The Connection_Handle shall be a Connection_Handle for an ACL connection. See [Section 6.18](#).

The default value is the value set by the HCI_Write_Default_Link_Policy_Settings command.

Command parameters:

Connection_Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xXXXX	Connection_Handle Range: 0x0000 to 0x0EFF

Link_Policy_Settings: *Size: 2 octets*

Bit Number	Parameter Description
0	Enable Role Switch.
1	Enable Hold mode.
2	Enable Sniff mode.
All other bits	Reserved for future use.

Return parameters:

Status: *Size: 1 octet*

Value	Parameter Description
0x00	HCI_Write_Link_Policy_Settings command succeeded.
0x01 to 0xFF	HCI_Write_Link_Policy_Settings command failed. See [Vol 1] Part F, Controller Error Codes for error codes and descriptions.

**Connection_Handle:****Size: 2 octets (12 bits meaningful)**

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

Event(s) generated (unless masked away):

When the HCI_Write_Link_Policy_Settings command has completed, an HCI_Command_Complete event shall be generated.



7.2.11 Read Default Link Policy Settings command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Default_Link_Policy_Settings	0x000E		Status, Default_Link_Policy_Settings

Description:

This command reads the Default Link Policy setting for all new BR/EDR connections.

Note: See the Link Policy Settings configuration parameter for more information. See [Section 6.18](#).

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Link_Policy_Settings command succeeded
0x01 to 0xFF	HCI_Read_Link_Policy_Settings command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Default_Link_Policy_Settings:

Size: 2 octets

Bit Number	Parameter Description
0	Enable Role Switch
1	Enable Hold mode
2	Enable Sniff mode
All other bits	Reserved for future use

Event(s) generated (unless masked away):

When the HCI_Read_Default_Link_Policy_Settings command has completed, an HCI_Command_Complete event shall be generated.



7.2.12 Write Default Link Policy Settings command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Default_Link_Policy_Settings	0x000F	Default_Link_Policy_Settings	Status

Description:

This command writes the Default Link Policy configuration value. The Default_Link_Policy_Settings parameter determines the initial value of the Link_Policy_Settings for all new BR/EDR connections.

Note: See the Link Policy Settings configuration parameter for more information. See [Section 6.18](#).

Command parameters:

Default_Link_Policy_Settings:

Size: 2 octets

Bit Number	Parameter Description
0	Enable Role Switch
1	Enable Hold mode
2	Enable Sniff mode
All other bits	Reserved for future use

The default value is 0x0000.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Write_Link_Policy_Settings command succeeded
0x01 to 0xFF	HCI_Write_Link_Policy_Settings command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

When the HCI_Write_Default_Link_Policy_Settings command has completed, an HCI_Command_Complete event will be generated.



7.2.13 Flow Specification command

Command	OCF	Command Parameters	Return Parameters
HCI_Flow_Specification	0x0010	Connection_Handle, Unused, Flow_Direction, Service_Type, Token_Rate, Token_Bucket_Size, Peak_Bandwidth, Access_Latency	

Description:

The HCI_Flow_Specification command is used to specify the flow parameters for the traffic carried over the ACL connection identified by the Connection_Handle. The Connection_Handle shall be a Connection_Handle for an ACL connection. The Connection_Handle command parameter is used to identify for which connection the Flow Specification is requested. The flow parameters refer to the outgoing or incoming traffic of the ACL link, as indicated by the Flow_Direction field. The HCI_Flow_Specification command allows the Link Manager to have the parameters of the outgoing as well as the incoming flow for the ACL connection. The flow parameters are defined in the L2CAP specification [\[Vol 3\] Part A, Section 5.3, Quality of Service \(QoS\) option](#). The Link Manager will determine if the flow parameters can be supported. BR/EDR Controllers that are both Central and Peripheral can use this command.

The Unused parameter is reserved for future use.

Command parameters:

Connection_Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xXXXX	Connection_Handle Range: 0x0000 to 0x0EFF

Unused: *Size: 1 octet*

Value	Parameter Description
0x00	This value shall be used by the Host.
All other values	Reserved for future use.

**Flow_Direction:****Size: 1 octet**

Value	Parameter Description
0x00	Outgoing Flow i.e., traffic sent over the ACL connection
0x01	Incoming Flow i.e., traffic received over the ACL connection
All other values	Reserved for future use.

Service_Type:**Size: 1 octet**

Value	Parameter Description
0x00	No Traffic
0x01	Best Effort
0x02	Guaranteed
All other values	Reserved for future use

Token_Rate:**Size: 4 octets**

Value	Parameter Description
0xFFFFFFFF	Token Rate in octets per second

Token_Bucket_Size:**Size: 4 octets**

Value	Parameter Description
0xFFFFFFFF	Token Bucket Size in octets

Peak_Bandwidth:**Size: 4 octets**

Value	Parameter Description
0xFFFFFFFF	Peak Bandwidth in octets per second

Access_Latency:**Size: 4 octets**

Value	Parameter Description
0xFFFFFFFF	Latency in microseconds

Return parameters:

None.

Event(s) generated (unless masked away):

When the Controller receives the HCI_Flow_Specification command, the BR/EDR Controller shall send the HCI_Command_Status event to the Host. When the Link Manager has determined if the Flow specification can be supported,



the BR/EDR Controller on the local BR/EDR Controller shall send an HCI_Flow_Specification_Complete event to the Host. The HCI_Flow_Specification_Complete event returned by the Controller on the local side contains the status of this command, and returned Flow parameters describing the supported QoS for the ACL connection.

Note: If the Link Manager performs an LMP transaction that involves the flow parameter values on the remote side, the remote Controller can send an HCI_Flow_Specification_Complete event or HCI_QoS_Setup_Complete event to the remote Host.



7.2.14 Sniff Subrating command

Command	OCF	Command Parameters	Return Parameters
HCI_Sniff_Subrating	0x0011	Connection_Handle, Max_Latency, Min_Remote_Timeout, Min_Local_Timeout	Status, Connection_Handle

Description:

The HCI_Sniff_Subrating command specifies the parameters for sniff subrating for a given link. The interval shall be determined from the sniff interval and the maximum subrate latency parameters from the command. The link may have smaller subrates and therefore lower latencies and longer timeouts than those specified. When the sniff subrate has been exchanged an HCI_Sniff_Subrating event shall be generated. If this command is used on a link in Sniff mode this shall cause sniff subrating to be negotiated at the Link Manager, otherwise sniff subrating shall be negotiated only after the device has entered Sniff mode.

The Connection_Handle shall be the primary Connection_Handle between the two devices.

The Maximum Latency parameter shall define the maximum allowed sniff subrate of the remote device.

If the Host does not write the sniff subrating parameters prior to sniff subrating being initiated by the Link Manager the default values shall be used.

Command parameters:

Connection_Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xXXXX	Connection_Handle Range: 0x0000 to 0x0EFF

Max_Latency: *Size: 2 octets*

Value	Parameter Description
N = 0xXXXX	The Maximum Latency parameter shall be used to calculate the maximum_sniff subrate that the remote device may use. Default: T_{sniff} Latency = $N * 0.625$ ms (1 Baseband slot) Range: 0x0002 to 0xFFFFE Time Range: 1.25 ms to 40.9 s

**Min_Remote_Timeout:****Size: 2 octets**

Value	Parameter Description
N = 0xFFFF	Minimum sniff mode timeout ($T_{\text{sniff_mode_timeout}}$) that the remote device may use Default: 0x0000 Timeout = $N * 0.625 \text{ ms}$ (1 Baseband slot) Range: 0x0000 to 0xFFFF Time Range: 0 s to 40.9 s

Min_Local_Timeout:**Size: 2 octets**

Value	Parameter Description
N = 0xFFFF	Minimum sniff mode timeout ($T_{\text{sniff_mode_timeout}}$) that the local device may use. Default: 0x0000 Timeout = $N * 0.625 \text{ ms}$ (1 Baseband slot) Range: 0x0000 to 0xFFFF Time Range: 0 s to 40.9 s

Return parameters:**Status:****Size: 1 octet**

Value	Parameter Description
0x00	The HCI_Sniff_Subrating command succeeded.
0x01 to 0xFF	HCI_Sniff_Subrating command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Connection_Handle:**Size: 2 octets (12 bits meaningful)**

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

Event(s) generated (unless masked away):

When the HCI_Sniff_Subrating command has been received by the BR/EDR Controller, an HCI_Command_Complete event shall be generated.

An HCI_Sniff_Subrating event shall occur when the sniff subrating has been negotiated for the specified Connection_Handle.



7.3 CONTROLLER & BASEBAND COMMANDS

The Controller & Baseband commands provide access and control to various capabilities of the Bluetooth hardware. These parameters provide control of BR/EDR Controllers and of the capabilities of the Link Manager and Baseband in the BR/EDR Controller and the Link Layer in an LE Controller. The Host can use these commands to modify the behavior of the local Controller.

For the HCI Control and Baseband commands, the OGF is defined as 0x03.

7.3.1 Set Event Mask command

Command	OCF	Command Parameters	Return Parameters
HCI_Set_Event_Mask	0x0001	Event_Mask	Status

Description:

The HCI_Set_Event_Mask command is used to control which events are generated by the HCI for the Host. If the bit in the Event_Mask is set to a one, then the event associated with that bit will be enabled. For an LE Controller, the “LE Meta event” bit in the event_mask shall enable or disable all LE events in the LE Meta event (see [Section 7.7.65](#)). The event mask allows the Host to control how much it is interrupted.

The Controller shall ignore those bits which are reserved for future use or represent events which it does not support. If the Host sets any of these bits to 1, the Controller shall act as if they were set to 0.

Command parameters:

Event_Mask:

Size: 8 octets

Bit	Parameter Description
0	Inquiry Complete event
1	Inquiry Result event
2	Connection Complete event
3	Connection Request event
4	Disconnection Complete event
5	Authentication Complete event
6	Remote Name Request Complete event
7	Encryption Change event [v1]
8	Change Connection Link Key Complete event
9	Link Key Type Changed event



Bit	Parameter Description
10	Read Remote Supported Features Complete event
11	Read Remote Version Information Complete event
12	QoS Setup Complete event
15	Hardware Error event
16	Flush Occurred event
17	Role Change event
19	Mode Change event
20	Return Link Keys event
21	PIN Code Request event
22	Link Key Request event
23	Link Key Notification event
24	Loopback Command event
25	Data Buffer Overflow event
26	Max Slots Change event
27	Read Clock Offset Complete event
28	Connection Packet Type Changed event
29	QoS Violation event
30	Previously used
31	Page Scan Repetition Mode Change event
32	Flow Specification Complete event
33	Inquiry Result with RSSI event
34	Read Remote Extended Features Complete event
43	Synchronous Connection Complete event
44	Synchronous Connection Changed event
45	Sniff Subrating event
46	Extended Inquiry Result event
47	Encryption Key Refresh Complete event
48	IO Capability Request event
49	IO Capability Response event
50	User Confirmation Request event
51	User Passkey Request event
52	Remote OOB Data Request event



Bit	Parameter Description
53	Simple Pairing Complete event
55	Link Supervision Timeout Changed event
56	Enhanced Flush Complete event
58	User Passkey Notification event
59	Keypress Notification event
60	Remote Host Supported Features Notification event
61	LE Meta event
All other bits	Reserved for future use

The value with all bits set to 0 indicates that no events are specified.
The default is for bits 0 to 44 (the value 0x0000 1FFF FFFF FFFF) to be set.

Return parameters:*Status:**Size: 1 octet*

Value	Parameter Description
0x00	HCI_Set_Event_Mask command succeeded.
0x01 to 0xFF	HCI_Set_Event_Mask command failed. See [Vol 1] Part F, Controller Error Codes for error codes and descriptions.

Event(s) generated (unless masked away):

When the HCI_Set_Event_Mask command has completed, an HCI_Command_Complete event shall be generated.



7.3.2 Reset command

Command	OCF	Command Parameters	Return Parameters
HCI_Reset	0x0003		Status

Description:

The HCI_Reset command will reset the Controller and the Link Manager on the BR/EDR Controller or the Link Layer on an LE Controller. If the Controller supports both BR/EDR and LE then the HCI_Reset command shall reset the Link Manager, Baseband and Link Layer. The HCI_Reset command shall not affect the used HCI transport layer since the HCI transport layers may have reset mechanisms of their own. After the reset is completed, the current operational state will be lost, the Controller will enter standby mode and the Controller will automatically revert to the default values for the parameters for which default values are defined in the specification.

Note: The HCI_Reset command will not necessarily perform a hardware reset. This is implementation defined.

The Host shall not send additional HCI commands before the HCI_Command_Complete event related to the HCI_Reset command has been received.

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Reset command succeeded, was received and will be executed.
0x01 to 0xFF	HCI_Reset command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

When the reset has been performed, an HCI_Command_Complete event shall be generated.



7.3.3 Set Event Filter command

Command	OCF	Command Parameters	Return Parameters
HCI_Set_Event_Filter	0x0005	Filter_Type, Filter_Condition_Type, Condition	Status

Description:

The HCI_Set_Event_Filter command is used by the Host to specify different event filters. The Host may issue this command multiple times to request various conditions for the same type of event filter and for different types of event filters. The event filters are used by the Host to specify items of interest, which allow the BR/EDR Controller to send only events which interest the Host. Only some of the events have event filters. By default (before this command has been issued after power-on or Reset) no filters are set, and the Auto_Accept_Flag is off (incoming connections are not automatically accepted). An event filter is added each time this command is sent from the Host and the Filter_Condition_Type is not equal to 0x00. (The old event filters will not be overwritten). To clear all event filters, the Filter_Type = 0x00 is used. The Auto_Accept_Flag will then be set to off. To clear event filters for only a certain Filter_Type, the Filter_Condition_Type = 0x00 is used.

The Inquiry Result filter allows the BR/EDR Controller to filter out HCI_Inquiry_Result, HCI_Inquiry_Result_with_RSSI, or HCI_Extended_Inquiry_Result events. The Inquiry Result filter allows the Host to specify that the BR/EDR Controller only sends Inquiry Results to the Host if the report meets one of the specified conditions set by the Host. For the Inquiry Result filter, the Host can specify one or more of the following Filter Condition Types:

1. Return responses from all devices during the Inquiry process
2. A device with a specific Class of Device responded to the Inquiry process
3. A device with a specific BD_ADDR responded to the Inquiry process

The Inquiry Result filter is used in conjunction with the HCI_Inquiry and HCI_Periodic_Inquiry_Mode commands.

The Connection Setup filter allows the Host to specify that the Controller only sends an HCI_Connection_Complete, HCI_Synchronous_Connection_Complete, or HCI_Connection_Request event to the Host if the event meets one of the specified conditions set by the Host. For the Connection Setup filter, the Host can specify one or more of the following Filter Condition Types:

1. Allow Connections from all devices
2. Allow Connections from a device with a specific Class of Device
3. Allow Connections from a device with a specific BD_ADDR



For each of these conditions, an `Auto_Accept_Flag` parameter allows the Host to specify what action should be done when the condition is met. The `Auto_Accept_Flag` allows the Host to specify if the incoming connection should be auto accepted (in which case the BR/EDR Controller will send a `HCI_Connection_Complete` event to the Host when an ACL or SCO connection is completed or an `HCI_Synchronous_Connection_Complete` event when an eSCO connection is completed) or if the Host should make the decision (in which case the BR/EDR Controller will send the `HCI_Connection_Request` event to the Host, to elicit a decision on the connection). If the `Auto_Accept_Flag` is off and the Host has masked the `HCI_Connection_Request` event, the Controller shall reject the connection attempt.

Note: Auto-accept does not override any requirement to reject a connection in this specification, such as the requirement in [\[Vol 2\] Part C, Section 4.5.1](#) to reject a SCO connection when AES-CCM encryption is in use.

If a synchronous connection is auto-accepted, then the default parameter settings of the `Accept_Synchronous_Connection_Request` command (see [Section 7.1.27](#)) should be used by the local Link Manager when negotiating the SCO or eSCO link parameters.

The Connection Setup filter is used in conjunction with the `HCI_Read/Write_Scan_Enable` commands. If the local device is in the process of a page scan, and is paged by another device which meets one of the conditions set by the Host, and the `Auto_Accept_Flag` is off for this device, then an `HCI_Connection_Request` event will be sent to the Host by the BR/EDR Controller. An `HCI_Connection_Complete` event will be sent later on after the Host has responded to the incoming connection attempt. In this same example, if the `Auto_Accept_Flag` is on, then an `HCI_Connection_Complete` event will be sent to the Host by the Controller. (No `HCI_Connection_Request` event will be sent in that case.)

The BR/EDR Controller will store these filters in volatile memory until the Host clears the event filters using the `HCI_Set_Event_Filter` command or until the `HCI_Reset` command is issued. The number of event filters the BR/EDR Controller can store is implementation dependent. If the Host tries to set more filters than the BR/EDR Controller can store, the BR/EDR Controller will return the error code *Memory Capacity Exceeded* (0x07) and the filter will not be installed.

Note: The Clear All Filters has no Filter Condition Types or Conditions.

Note: In the condition that a connection is auto accepted, an `HCI_Link_Key_Request` event and possibly also an `HCI_PIN_Code_Request` event and an `HCI_Link_Key_Notification` event could be sent to the Host by the Controller before the `HCI_Connection_Complete` event is sent.

If there is a contradiction between event filters, the latest set event filter will override older ones. An example is an incoming connection attempt where



more than one Connection Setup filter matches the incoming connection attempt, but the Auto-Accept_Flag has different values in the different filters.

Command parameters:*Filter_Type:**Size: 1 octet*

Value	Parameter Description
0x00	Clear All Filters Note: In this case, the Filter_Condition_Type and Condition parameters should not be given; they should have a length of 0 octets. Filter_Type should be the only parameter.
0x01	Inquiry Result.
0x02	Connection Setup.
All other values	Reserved for future use.

Filter Condition Types: For each Filter Type one or more Filter Condition types exists.

*Inquiry_Result_Filter_Condition_Type:**Size: 1 octet*

Value	Parameter Description
0x00	Return responses from all devices during the Inquiry process. Note: A device may be reported to the Host in an HCI_Inquiry_Result, HCI_Inquiry_Result_with_RSSI, or HCI_Extended_Inquiry_Result event more than once during an inquiry or inquiry period depending on the implementation; see description in Section 7.1.1 and Section 7.1.3 .
0x01	A device with a specific Class of Device responded to the Inquiry process.
0x02	A device with a specific BD_ADDR responded to the Inquiry process.
All other values	Reserved for future use

*Connection_Setup_Filter_Condition_Type:**Size: 1 octet*

Value	Parameter Description
0x00	Allow Connections from all devices.
0x01	Allow Connections from a device with a specific Class of Device.
0x02	Allow Connections from a device with a specific BD_ADDR.
All other values	Reserved for future use.



Condition: For each Filter Condition Type defined for the Inquiry Result Filter and the Connection Setup Filter, zero or more Condition parameters are required – depending on the filter condition type and filter type.

Condition for Inquiry_Result_Filter_Condition_Type = 0x00

Condition:

Size: 0 octet

Value	Parameter Description
	The Condition parameter is not used.

Condition for Inquiry_Result_Filter_Condition_Type = 0x01

Condition:

Size: 6 octets

Class_Of_Device:

Size: 3 octets

Value	Parameter Description
0x000000	Default, Return All Devices.
0xxxxxxx	Class of Device of Interest.

Class_Of_Device_Mask:

Size: 3 octets

Value	Parameter Description
0xxxxxxx	Bit Mask used to determine which bits of the Class of Device parameter are 'don't care'. Zero-value bits in the mask indicate the 'don't care' bits of the Class of Device.

Condition for Inquiry_Result_Filter_Condition_Type = 0x02

Condition:

Size: 6 octets

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0xxxxxxxxxxx xx	BD_ADDR of the Device of Interest

Condition for Connection_Setup_Filter_Condition_Type = 0x00

Condition:

Size: 1 octet

Auto_Accept_Flag:

Size: 1 octet

Value	Parameter Description
0x01	Do NOT Auto accept the connection. (Auto accept is off)
0x02	Do Auto accept the connection with role switch disabled. (Auto accept is on).



Value	Parameter Description
0x03	Do Auto accept the connection with role switch enabled. (Auto accept is on). Note: When auto accepting an incoming synchronous connection, no role switch will be performed. The value 0x03 of the Auto_Accept_Flag will then get the same effect as if the value had been 0x02.
All other values	Reserved for future use.

Condition for Connection_Setup_Filter_Condition_Type = 0x01

Condition:

Size: 7 octets

Class_Of_Device:

Size: 3 octets

Value	Parameter Description
0x000000	Default, Return All Devices.
0xxxxxxx	<i>Class of Device</i> of Interest.

Class_Of_Device_Mask:

Size: 3 octets

Value	Parameter Description
0xxxxxxx	Bit Mask used to determine which bits of the Class of Device parameter are 'don't care'. Zero-value bits in the mask indicate the 'don't care' bits of the Class of Device. Note: For an incoming SCO connection, if the Class of Device is unknown then the connection will be accepted.

Auto_Accept_Flag:

Size: 1 octet

Value	Parameter Description
0x01	Do NOT Auto accept the connection. (Auto accept is off)
0x02	Do Auto accept the connection with role switch disabled. (Auto accept is on).
0x03	Do Auto accept the connection with role switch enabled. (Auto accept is on). Note: When auto accepting an incoming synchronous connection, no role switch will be performed. The value 0x03 of the Auto_Accept_Flag will then get the same effect as if the value had been 0x02.
All other values	Reserved for future use.



Condition for Connection_Setup_Filter_Condition_Type = 0x02

Condition:

Size: 7 octets

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0XXXXXXXXXX XX	BD_ADDR of the Device of Interest.

Auto_Accept_Flag:

Size: 1 octet

Value	Parameter Description
0x01	Do NOT Auto accept the connection. (Auto accept is off)
0x02	Do Auto accept the connection with role switch disabled. (Auto accept is on).
0x03	Do Auto accept the connection with role switch enabled. (Auto accept is on). Note: When auto accepting an incoming synchronous connection, no role switch will be performed. The value 0x03 of the Auto_Accept_Flag will then get the same effect as if the value had been 0x02.
All other values	Reserved for future use.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Set_Event_Filter command succeeded.
0x01 to 0xFF	HCI_Set_Event_Filter command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

An HCI_Command_Complete event for this command shall occur when the Controller has enabled the filtering of events. When one of the conditions are met, a specific event shall occur.



7.3.4 Flush command

Command	OCF	Command Parameters	Return Parameters
HCI_Flush	0x0008	Connection_Handle	Status, Connection_Handle

Description:

The HCI_Flush command is used to discard all data that is currently pending for transmission in the Controller for the specified Connection_Handle, even if there currently are chunks of data that belong to more than one L2CAP packet in the Controller. Both automatically-flushable and non-automatically-flushable packets shall be discarded (see [Section 5.4.2](#)). After this, all data that is sent to the Controller for the same Connection_Handle will be discarded by the Controller until an HCI ACL Data packet with one of the start Packet_Boundary_Flag values (0x00 or 0x02) is received. When this happens, a new transmission attempt can be made.

This command, when used on a BR/EDR Controller, will allow higher-level software to control how long the Baseband should try to retransmit a Baseband packet for a Connection_Handle before all data that is currently pending for transmission in the Controller should be flushed.

Note: The HCI_Flush command is used for ACL connections only. In addition to the HCI_Flush command, the automatic flush timers (see [Section 7.3.29](#)) can be used to automatically flush an automatically-flushable L2CAP packet that is currently being transmitted after the specified flush timer has expired.

Command parameters:

Connection_Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xXXXX	Connection_Handle Range: 0x0000 to 0x0EFF

Return parameters:

Status: *Size: 1 octet*

Value	Parameter Description
0x00	HCI_Flush command succeeded.
0x01 to 0xFF	HCI_Flush command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.



Connection_Handle: **Size: 2 octets (12 bits meaningful)**

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0xFFFF

Event(s) generated (unless masked away):

The HCI_Flush_Occurred event shall occur once the flush is completed. An HCI_Flush_Occurred event could be from an automatic Flush or could be caused by the Host issuing the HCI_Flush command. When the HCI_Flush command has completed, an HCI_Command_Complete event shall be generated, to indicate that the Host caused the Flush.



7.3.5 Read PIN Type command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_PIN_Type	0x0009		Status, PIN_Type

Description:

The HCI_Read_PIN_Type command is used to read the PIN_Type configuration parameter. See [Section 6.13](#).

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_PIN_Type command succeeded.
0x01 to 0xFF	HCI_Read_PIN_Type command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

PIN_Type:

Size: 1 octet

Value	Parameter Description
0x00	Variable PIN.
0x01	Fixed PIN.

Event(s) generated (unless masked away):

When the HCI_Read_PIN_Type command has completed, an HCI_Command_Complete event will be generated.



7.3.6 Write PIN Type command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_PIN_Type	0x000A	PIN_Type	Status

Description:

The HCI_Write_PIN_Type command is used to write the PIN Type configuration parameter. See [Section 6.13](#).

Command parameters:

PIN_Type:

Size: 1 octet

Value	Parameter Description
0x00	Variable PIN.
0x01	Fixed PIN.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Write_PIN_Type command succeeded.
0x01 to 0xFF	HCI_Write_PIN_Type command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

When the HCI_Write_PIN_Type command has completed, an HCI_Command_Complete event shall be generated.



7.3.7 [This section is no longer used]



7.3.8 Read Stored Link Key command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Stored_Link_Key	0x000D	BD_ADDR, Read_All	Status, Max_Num_Keys, Num_Keys_Read

Description:

The HCI_Read_Stored_Link_Key command provides the ability to read whether one or more link keys are stored in the BR/EDR Controller. The BR/EDR Controller can store a limited number of link keys for other BR/EDR Controllers. Link keys are shared between two BR/EDR Controllers, and are used for all security transactions between the two devices. The HCI_Read_Stored_Link_Key command shall not return the link key's value. A Host device may have additional storage capabilities, which can be used to save additional link keys to be reloaded to the BR/EDR Controller when needed. The Read_All parameter is used to indicate if all of the stored Link Keys should be returned. If Read_All indicates that all Link Keys are to be returned, then the BD_ADDR command parameter shall be ignored. The BD_ADDR command parameter is used to identify which link key to read. The stored Link Keys are returned by one or more HCI_Return_Link_Keys events. See [Section 6.14](#).

Command parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0XXXXXXXXXXXXX	BD_ADDR for the stored link key to be read.

Read_All:

Size: 1 octet

Value	Parameter Description
0x00	Return Link Key for specified BD_ADDR.
0x01	Return all stored Link Keys.
All other values	Reserved for future use.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Stored_Link_Key command succeeded.
0x01 to 0xFF	HCI_Read_Stored_Link_Key command failed. See [Vol 1] Part F, Controller Error Codes for error codes and descriptions.

**Max_Num_Keys:****Size: 2 octets**

Value	Parameter Description
0xFFFF	Total Number of Link Keys that the Controller can store. Range: 0x0000 to 0xFFFF

Num_Keys_Read:**Size: 2 octets**

Value	Parameter Description
0xFFFF	Number of Link Keys Read. Range: 0x0000 to 0xFFFF

Event(s) generated (unless masked away):

Zero or more instances of the HCI_Return_Link_Keys event shall occur after the command is issued. When there are no link keys stored, no HCI_Return_Link_Keys events will be returned. When there are link keys stored, the number of link keys returned in each HCI_Return_Link_Keys event is implementation specific. When the HCI_Read_Stored_Link_Key command has completed an HCI_Command_Complete event shall be generated.



7.3.9 Write Stored Link Key command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Stored_Link_Key	0x0011	Num_Keys_To_Write, BD_ADDR[i], Link_Key[i]	Status, Num_Keys_Written

Description:

The HCI_Write_Stored_Link_Key command provides the ability to write one or more link keys to be stored in the BR/EDR Controller. The BR/EDR Controller can store a limited number of link keys for other BR/EDR Controllers. If no additional space is available in the BR/EDR Controller then no additional link keys will be stored. If space is limited and if all the link keys to be stored will not fit in the limited space, then the order of the list of link keys without any error will determine which link keys are stored. Link keys at the beginning of the list will be stored first. The Num_Keys_Written parameter will return the number of link keys that were successfully stored. If no additional space is available, then the Host must delete one or more stored link keys before any additional link keys are stored. The link key replacement algorithm is implemented by the Host and not the BR/EDR Controller. Link keys are shared between two BR/EDR Controllers and are used for all security transactions between the two devices. A Host device may have additional storage capabilities, which can be used to save additional link keys to be reloaded to the BR/EDR Controller when needed. See [Section 6.14](#).

Note: Link Keys are only stored by issuing this command.

A Host in Secure Connections Only Mode should not store link keys in the Controller.

Command parameters:

Num_Keys_To_Write:

Size: 1 octet

Value	Parameter Description
0xXX	Number of Link Keys to Write. Range: 0x01 to 0x0B

BD_ADDR[i]:

Size: Num_Keys_To_Write × 6 octets

Value	Parameter Description
0XXXXXXXXXXXXX	BD_ADDR for the associated Link Key.

*Link_Key[i]:**Size: Num_Keys_To_Write × 16 octets*

Value	Parameter Description
0xFFFFFFFFXXXXXXX XXXXXXXXXXXX XXXXXXXXXXXX	Link Key for an associated BD_ADDR.

Return parameters:*Status:**Size: 1 octet*

Value	Parameter Description
0x00	HCI_Write_Stored_Link_Key command succeeded.
0x01 to 0xFF	HCI_Write_Stored_Link_Key command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

*Num_Keys_Written:**Size: 1 octet*

Value	Parameter Description
0xXX	Number of Link Keys successfully written. Range: 0x00 to 0x0B

Event(s) generated (unless masked away):

When the HCI_Write_Stored_Link_Key command has completed, an HCI_Command_Complete event shall be generated.



7.3.10 Delete Stored Link Key command

Command	OCF	Command Parameters	Return Parameters
HCI_Delete_Stored_Link_Key	0x0012	BD_ADDR, Delete_All	Status, Num_Keys_Deleted

Description:

The HCI_Delete_Stored_Link_Key command provides the ability to remove one or more of the link keys stored in the BR/EDR Controller. The BR/EDR Controller can store a limited number of link keys for other BR/EDR devices. Link keys are shared between two BR/EDR devices and are used for all security transactions between the two devices. The Delete_All parameter is used to indicate if all of the stored Link Keys should be deleted. If the Delete_All indicates that all Link Keys are to be deleted, then the BD_ADDR command parameter shall be ignored. This command provides the ability to negate all security agreements between two devices. The BD_ADDR command parameter is used to identify which link key to delete. If a link key is currently in use for a connection, then the link key will be deleted when all of the connections are disconnected. See [Section 6.14](#).

Command parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0XXXXXXXXXXXXX	BD_ADDR for the link key to be deleted.

Delete_All:

Size: 1 octet

Value	Parameter Description
0x00	Delete only the Link Key for specified BD_ADDR.
0x01	Delete all stored Link Keys.
All other values	Reserved for future use.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Delete_Stored_Link_Key command succeeded.
0x01 to 0xFF	HCI_Delete_Stored_Link_Key command failed. See [Vol 1] Part F, Controller Error Codes for error codes and descriptions.



Num_Keys_Deleted:

Size: 2 octets

Value	Parameter Description
0xFFFF	Number of Link Keys Deleted

Event(s) generated (unless masked away):

When the HCI_Delete_Stored_Link_Key command has completed, an HCI_Command_Complete event shall be generated.



7.3.11 Write Local Name command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Local_Name	0x0013	Local_Name	Status

Description:

The HCI_Write_Local_Name command provides the ability to modify the user-friendly name for the BR/EDR Controller. See [Section 6.23](#).

Command parameters:

Local_Name:

Size: 248 octets

Value	Parameter Description
	A UTF-8 encoded User-Friendly Descriptive Name for the device with type utf8{248} (see [Vol 1] Part E, Section 2.9.3).

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Write_Local_Name command succeeded.
0x01 to 0xFF	HCI_Write_Local_Name command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

When the HCI_Write_Local_Name command has completed, an HCI_Command_Complete event shall be generated.



7.3.12 Read Local Name command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Local_Name	0x0014		Status, Local_Name

Description:

The HCI_Read_Local_Name command provides the ability to read the stored user-friendly name for the BR/EDR Controller. See [Section 6.23](#).

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Local_Name command succeeded
0x01 to 0xFF	HCI_Read_Local_Name command failed (see [Vol 1] Part F, Controller Error Codes for list of error codes).

Local_Name:

Size: 248 octets

Value	Parameter Description
	A UTF-8 encoded User Friendly Descriptive Name for the device.with type utf8{248} (see [Vol 1] Part E, Section 2.9.3)

Event(s) generated (unless masked away):

When the HCI_Read_Local_Name command has completed an HCI_Command_Complete event shall be generated.



7.3.13 Read Connection Accept Timeout command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_ Connection_Accept_ Timeout	0x0015		Status, Connection_Accept_ Timeout

Description:

This command reads the value for the Connection_Accept_Timeout configuration parameter. See [Section 6.7](#).

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Connection_Accept_Timeout command succeeded.
0x01 to 0xFF	HCI_Read_Connection_Accept_Timeout command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Connection_Accept_Timeout:

Size: 2 octets

Value	Parameter Description
N = 0xXXXX	Connection Accept Timeout measured in number of BR/EDR Baseband slots. Interval Length = $N * 0.625 \text{ ms}$ (1 Baseband slot) Range: 0x0001 to 0xB540 Time Range: 0.625 ms to 29 s

Event(s) generated (unless masked away):

When the HCI_Read_Connection_Timeout command has completed, an HCI_Command_Complete event shall be generated.



7.3.14 Write Connection Accept Timeout command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_ - Connection_Accept_ - Timeout	0x0016	Connection_Accept_ - Timeout	Status

Description:

This command writes the value for the Connection_Accept_Timeout configuration parameter. See [Section 6.7](#).

Command parameters:

Connection_Accept_Timeout:

Size: 2 octets

Value	Parameter Description
N = 0xXXXX	Connection Accept Timeout measured in number of BR/EDR Baseband slots. Interval Length = $N * 0.625$ ms (1 Baseband slot) Range: 0x0001 to 0xB540 Time Range: 0.625 ms to 29 s Default: 0x1FA0 Time = 5.06 s Mandatory Range for Controller: 0x00A0 to 0xB540

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Write_Connection_Accept_Timeout command succeeded.
0x01 to 0xFF	HCI_Write_Connection_Accept_Timeout command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

When the HCI_Write_Connection_Accept_Timeout command has completed, an HCI_Command_Complete event shall be generated.



7.3.15 Read Page Timeout command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Page_Timeout	0x0017		Status, Page_Timeout

Description:

This command reads the value for the Page_Timeout configuration parameter. See [Section 6.6](#).

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Page_Timeout command succeeded.
0x01 to 0xFF	HCI_Read_Page_Timeout command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Page_Timeout:

Size: 2 octets

Value	Parameter Description
N = 0xFFFF	Page Timeout measured in number of Baseband slots. Interval Length = $N * 0.625 \text{ ms}$ (1 Baseband slot) Range: 0x0001 to 0xFFFF Time Range: 0.625 ms to 40.9 s

Event(s) generated (unless masked away):

When the HCI_Read_Page_Timeout command has completed, an HCI_Command_Complete event shall be generated.



7.3.16 Write Page Timeout command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Page_Timeout	0x0018	Page_Timeout	Status

Description:

This command writes the value for the Page_Timeout configuration parameter. See [Section 6.6](#).

Command parameters:

Page_Timeout:

Size: 2 octets

Value	Parameter Description
N = 0xXXXX	Page Timeout measured in number of Baseband slots. Interval Length = $N * 0.625$ ms (1 Baseband slot) Range: 0x0001 to 0xFFFF Time Range: 0.625 ms to 40.9 s Default: 0x2000 Time = 5.12 s Mandatory Range for Controller: 0x0016 to 0xFFFF

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Write_Page_Timeout command succeeded.
0x01 to 0xFF	HCI_Write_Page_Timeout command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

When the HCI_Write_Page_Timeout command has completed, an HCI_Command_Complete event shall be generated.



7.3.17 Read Scan Enable command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Scan_Enable	0x0019		Status, Scan_Enable

Description:

This command reads the value for the Scan_Enable parameter configuration parameter. See [Section 6.1](#).

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Scan_Enable command succeeded.
0x01 to 0xFF	HCI_Read_Scan_Enable command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Scan_Enable:

Size: 1 octet

Value	Parameter Description
0x00	No Scans enabled.
0x01	Inquiry Scan enabled. Page Scan disabled.
0x02	Inquiry Scan disabled. Page Scan enabled.
0x03	Inquiry Scan enabled. Page Scan enabled.
All other values	Reserved for future use

Event(s) generated (unless masked away):

When the HCI_Read_Scan_Enable command has completed, an HCI_Command_Complete event shall be generated.



7.3.18 Write Scan Enable command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Scan_Enable	0x001A	Scan_Enable	Status

Description:

This command writes the value for the Scan_Enable configuration parameter. See [Section 6.1](#).

Command parameters:

Scan_Enable:

Size: 1 octet

Value	Parameter Description
0x00	No Scans enabled. Default.
0x01	Inquiry Scan enabled. Page Scan disabled.
0x02	Inquiry Scan disabled. Page Scan enabled.
0x03	Inquiry Scan enabled. Page Scan enabled.
All other values	Reserved for future use

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Write_Scan_Enable command succeeded.
0x01 to 0xFF	HCI_Write_Scan_Enable command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

When the HCI_Write_Scan_Enable command has completed, an HCI_Command_Complete event shall be generated.



7.3.19 Read Page Scan Activity command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Page_Scan_Activity	0x001B		Status, Page_Scan_Interval, Page_Scan_Window

Description:

This command reads the value for Page_Scan_Interval and Page_Scan_Window configuration parameters. See [Section 6.8](#) and [Section 6.9](#).

Note: Page Scan is only performed when Page_Scan is enabled (see [6.1](#), [7.3.17](#) and [7.3.18](#)). A changed Page_Scan_Interval could change the local Page Scan Repetition Mode (see [\[Vol 2\] Part B, Section 8.3.1](#)).

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Page_Scan_Activity command succeeded.
0x01 to 0xFF	HCI_Read_Page_Scan_Activity command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Page_Scan_Interval:

Size: 2 octets

Value	Parameter Description
N = 0xXXXX	Range: 0x0012 to 0x1000 Time = N * 0.625 ms Range: 11.25 ms to 2560 ms; only even values are valid

Page_Scan_Window:

Size: 2 octets

Value	Parameter Description
N = 0xXXXX	Range: 0x0011 to 0x1000 Time = N * 0.625 ms Range: 10.625 ms to 2560 ms

**Event(s) generated (unless masked away):**

When the HCI_Read_Page_Scan_Activity command has completed, an HCI_Command_Complete event shall be generated.



7.3.20 Write Page Scan Activity command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Page_Scan_Activity	0x001C	Page_Scan_Interval, Page_Scan_Window	Status

Description:

This command writes the values for the Page_Scan_Interval and Page_Scan_Window configuration parameters. The Page_Scan_Window shall be less than or equal to the Page_Scan_Interval. See [Section 6.8](#) and [Section 6.9](#).

Note: Page Scan is only performed when Page_Scan is enabled (see [6.1](#), [7.3.17](#) and [7.3.18](#)). A changed Page_Scan_Interval could change the local Page Scan Repetition Mode (see [\[Vol 2\] Part B, Section 8.3.1](#)).

Command parameters:

Page_Scan_Interval:

Size: 2 octets

Value	Parameter Description
See Section 6.8	

Page_Scan_Window:

Size: 2 octets

Value	Parameter Description
See Section 6.9	

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Write_Page_Scan_Activity command succeeded.
0x01 to 0xFF	HCI_Write_Page_Scan_Activity command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

When the HCI_Write_Page_Scan_Activity command has completed, an HCI_Command_Complete event shall be generated.



7.3.21 Read Inquiry Scan Activity command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Inquiry_Scan_Activity	0x001D		Status, Inquiry_Scan_Interval, Inquiry_Scan_Window

Description:

This command reads the value for Inquiry_Scan_Interval and Inquiry_Scan_Window configuration parameter. See [Section 6.2](#) and [Section 6.3](#).

Note: Inquiry Scan is only performed when Inquiry_Scan is enabled see [6.1](#), [7.3.17](#) and [7.3.18](#)).

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Inquiry_Scan_Activity command succeeded.
0x01 to 0xFF	HCI_Read_Inquiry_Scan_Activity command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Inquiry_Scan_Interval:

Size: 2 octets

Value	Parameter Description
N = 0xXXXX	Range: 0x0012 to 0x1000 Time = N * 0.625 ms Range: 11.25 to 2560 ms; only even values are valid

Inquiry_Scan_Window:

Size: 2 octets

Value	Parameter Description
N = 0xXXXX	Range: 0x0011 to 0x1000 Time = N * 0.625 ms Range: 10.625 ms to 2560 ms

Event(s) generated (unless masked away):

When the HCI_Read_Inquiry_Scan_Activity command has completed, an HCI_Command_Complete event shall be generated.



7.3.22 Write Inquiry Scan Activity command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Inquiry_Scan_Activity	0x001E	Inquiry_Scan_Interval, Inquiry_Scan_Window	Status

Description:

This command writes the values for the Inquiry_Scan_Interval and Inquiry_Scan_Window configuration parameters. The Inquiry_Scan_Window shall be less than or equal to the Inquiry_Scan_Interval. See [Section 6.2](#) and [Section 6.3](#).

Note: Inquiry Scan is only performed when Inquiry_Scan is enabled (see [6.1](#), [7.3.17](#) and [7.3.18](#)).

Command parameters:

Inquiry_Scan_Interval:

Size: 2 octets

Value	Parameter Description
See Section 6.2 .	

Inquiry_Scan_Window:

Size: 2 octets

Value	Parameter Description
See Section 6.3 .	

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Write_Inquiry_Scan_Activity command succeeded.
0x01 to 0xFF	HCI_Write_Inquiry_Scan_Activity command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

When the HCI_Write_Inquiry_Scan_Activity command has completed, an HCI_Command_Complete event shall be generated.



7.3.23 Read Authentication Enable command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_ - Authentication_Enable	0x001F		Status, Authentication_Enable

Description:

This command reads the value for the Authentication_Enable configuration parameter. See [Section 6.16 Authentication Enable](#).

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Authentication_Enable command succeeded.
0x01 to 0xFF	HCI_Read_Authentication_Enable command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Authentication_Enable:

Size: 1 octet

Value	Parameter Description
0x00	Authentication not required.
0x01	Authentication required for all connections.
All other values	Reserved for future use

Event(s) generated (unless masked away):

When the HCI_Read_Authentication_Enable command has completed, an HCI_Command_Complete event shall be generated.



7.3.24 Write Authentication Enable command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_ - Authentication_Enable	0x0020	Authentication_Enable	Status

Description:

This command writes the value for the Authentication_Enable configuration parameter. See [Section 6.16 Authentication Enable](#).

The Authentication_Enable configuration parameter shall only apply to connections (e.g. send an LMP_IN_RAND or LMP_AU_RAND) when the remote device's Host or BR/EDR Controller does not support Secure Simple Pairing or when the local Host does not support Secure Simple Pairing.

Note: Requires LM to read Host features during connection setup.

Command parameters:

Authentication_Enable:

Size: 1 octet

Value	Parameter Description
0x00	Authentication not required. Default.
0x01	Authentication required for all connections.
All other values	Reserved for future use

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Write_Authentication_Enable command succeeded.
0x01 to 0xFF	HCI_Write_Authentication_Enable command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

When the HCI_Write_Authentication_Enable command has completed, an HCI_Command_Complete event shall be generated.



7.3.25 Read Class of Device command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Class_Of_Device	0x0023		Status, Class_Of_Device

Description:

This command reads the value for the Class_Of_Device parameter. See [Section 6.26](#).

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Class_Of_Device command succeeded.
0x01 to 0xFF	HCI_Read_Class_Of_Device command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Class_Of_Device:

Size: 3 octets

Value	Parameter Description
0xXXXXXX	Class of Device for the device.

Event(s) generated (unless masked away):

When the HCI_Read_Class_Of_Device command has completed, an HCI_Command_Complete event shall be generated.



7.3.26 Write Class of Device command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Class_Of_Device	0x0024	Class_Of_Device	Status

Description:

This command writes the value for the Class_Of_Device parameter.
See [Section 6.26](#).

Command parameters:

Class_Of_Device:

Size: 3 octets

Value	Parameter Description
0xXXXXXX	Class of Device for the device.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Write_Class_Of_Device command succeeded.
0x01 to 0xFF	HCI_Write_Class_Of_Device command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

When the HCI_Write_Class_Of_Device command has completed, an HCI_Command_Complete event shall be generated.



7.3.27 Read Voice Setting command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Voice_Setting	0x0025		Status, Voice_Setting

Description:

This command reads the values for the Voice_Setting configuration parameter. See [Section 6.12](#).

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Voice_Setting command succeeded.
0x01 to 0xFF	HCI_Read_Voice_Setting command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Voice_Setting:

Size: 2 octets (10 bits meaningful)

Value	Parameter Description
See Section 6.12 .	

Event(s) generated (unless masked away):

When the HCI_Read_Voice_Setting command has completed, an HCI_Command_Complete event shall be generated.



7.3.28 Write Voice Setting command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Voice_Setting	0x0026	Voice_Setting	Status

Description:

This command writes the values for the Voice_Setting configuration parameter. See [Section 6.12](#).

Command parameters:

Voice_Setting: *Size: 2 octets (10 bits meaningful)*

Value	Parameter Description
See Section 6.12 .	

Return parameters:

Status: *Size: 1 octet*

Value	Parameter Description
0x00	HCI_Write_Voice_Setting command succeeded.
0x01 to 0xFF	HCI_Write_Voice_Setting command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

When the HCI_Write_Voice_Setting command has completed, an HCI_Command_Complete event shall be generated.



7.3.29 Read Automatic Flush Timeout command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Automatic_Flush_Timeout	0x0027	Connection_Handle	Status, Connection_Handle, Flush_Timeout

Description:

This command reads the value for the Flush_Timeout parameter for the specified Connection_Handle. See [Section 6.19](#).

Command parameters:

Connection_Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Automatic_Flush_Timeout command succeeded.
0x01 to 0xFF	HCI_Read_Automatic_Flush_Timeout command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Connection_Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

Flush_Timeout:

Size: 2 octets

Value	Parameter Description
0x0000	Timeout = ∞ ; No Automatic Flush
N = 0xFFFF	Flush Timeout = N * 0.625 ms Range: 0x0001 to 0x07FF

Event(s) generated (unless masked away):

When the HCI_Read_Automatic_Flush_Timeout command has completed, an HCI_Command_Complete event shall be generated.



7.3.30 Write Automatic Flush Timeout command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Automatic_Flush_Timeout	0x0028	Connection_Handle, Flush_Timeout	Status, Connection_Handle

Description:

This command writes the value for the Flush_Timeout parameter for the specified Connection_Handle. See [Section 6.19](#).

Command parameters:

Connection_Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xXXXX	Connection_Handle Range: 0x0000 to 0x0EFF

Flush_Timeout: *Size: 2 octets*

Value	Parameter Description
0x0000	Timeout = ∞ ; No Automatic Flush. Default.
N = 0xXXXX	Flush Timeout = $N * 0.625$ ms Range: 0x0001 to 0x07FF Mandatory Range for Controller: 0x0002 to 0x07FF

Return parameters:

Status: *Size: 1 octet*

Value	Parameter Description
0x00	HCI_Write_Automatic_Flush_Timeout command succeeded.
0x01 to 0xFF	HCI_Write_Automatic_Flush_Timeout command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

**Connection_Handle:****Size: 2 octets (12 bits meaningful)**

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

Event(s) generated (unless masked away):

When the HCI_Write_Automatic_Flush_Timeout command has completed, an HCI_Command_Complete event shall be generated.



7.3.31 Read Num Broadcast Retransmissions command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Num_Broadcast_Retransmissions	0x0029		Status, Num_Broadcast_Retransmissions

Description:

This command reads the device's parameter value for the Number of Broadcast Retransmissions. See [Section 6.20](#)

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Num_Broadcast_Retransmissions command succeeded.
0x01 to 0xFF	HCI_Read_Num_Broadcast_Retransmissions command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Num_Broadcast_Retransmissions:

Size: 1 octet

Value	Parameter Description
See Section 6.20 .	

Event(s) generated (unless masked away):

When the HCI_Read_Num_Broadcast_Retransmission command has completed, an HCI_Command_Complete event shall be generated.



7.3.32 Write Num Broadcast Retransmissions command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Num_Broadcast_Retransmissions	0x002A	Num_Broadcast_Re-transmissions	Status

Description:

This command writes the device's parameter value for the Number of Broadcast Retransmissions. See [Section 6.20](#).

Command parameters:

Num_Broadcast_Retransmissions: *Size: 1 octet*

Value	Parameter Description
See Section 6.20 .	

Return parameters:

Status: *Size: 1 octet*

Value	Parameter Description
0x00	HCI_Write_Num_Broadcast_Retransmissions command succeeded.
0x01 to 0xFF	HCI_Write_Num_Broadcast_Retransmissions command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

When the HCI_Write_Num_Broadcast_Retransmissions command has completed, an HCI_Command_Complete event shall be generated.



7.3.33 Read Hold Mode Activity command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Hold_Mode_Activity	0x002B		Status, Hold_Mode_Activity

Description:

This command reads the value for the Hold_Mode_Activity parameter.
See [Section 6.16](#).

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Hold_Mode_Activity command succeeded.
0x01 to 0xFF	HCI_Read_Hold_Mode_Activity command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Hold_Mode_Activity:

Size: 1 octet

Value	Parameter Description
0x00	Maintain current Power State.
0x01	Suspend Page Scan.
0x02	Suspend Inquiry Scan.
0x04	Suspend Periodic Inquiries.
All other values	Reserved for future use.

Event(s) generated (unless masked away):

When the HCI_Read_Hold_Mode_Activity command has completed, an HCI_Command_Complete event shall be generated.



7.3.34 Write Hold Mode Activity command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Hold_Mode_Activity	0x002C	Hold_Mode_Activity	Status

Description:

This command writes the value for the Hold_Mode_Activity parameter.
See [Section 6.16](#).

Command parameters:

Hold_Mode_Activity:

Size: 1 octet

Value	Parameter Description
0x00	Maintain current Power State. Default.
0x01	Suspend Page Scan.
0x02	Suspend Inquiry Scan.
0x04	Suspend Periodic Inquiries.
All other values	Reserved for future use.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Write_Hold_Mode_Activity command succeeded.
0x01 to 0xFF	HCI_Write_Hold_Mode_Activity command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

When the HCI_Write_Hold_Mode_Activity command has completed, an HCI_Command_Complete event shall be generated.



7.3.35 Read Transmit Power Level command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Transmit_Power_Level	0x002D	Connection_Handle, Type	Status, Connection_Handle, TX_Power_Level

Description:

This command reads the values for the TX_Power_Level parameter for the specified Connection_Handle. The Connection_Handle shall be a Connection_Handle for an ACL connection.

Command parameters:

Connection_Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

Type: *Size: 1 octet*

Value	Parameter Description
0x00	Read Current Transmit Power Level.
0x01	Read Maximum Transmit Power Level.
All other values	Reserved for future use

Return parameters:

Status: *Size: 1 octet*

Value	Parameter Description
0x00	HCI_Read_Transmit_Power_Level command succeeded.
0x01 to 0xFF	HCI_Read_Transmit_Power_Level command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Connection_Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

**TX_Power_Level:****Size: 1 octet**

Value	Parameter Description
0xXX	Range: -30 to 20 Units: dBm

Event(s) generated (unless masked away):

When the HCI_Read_Transmit_Power_Level command has completed, an HCI_Command_Complete event shall be generated.



7.3.36 Read Synchronous Flow Control Enable command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_- Synchronous_Flow_- Control_Enable	0x002E		Status, Synchronous_Flow_Con- trol_Enable

Description:

The HCI_Read_Synchronous_Flow_Control_Enable command provides the ability to read the Synchronous_Flow_Control_Enable parameter. See [Section 6.22](#).

The Synchronous_Flow_Control_Enable parameter shall only be changed if no connection exists.

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Synchronous_Flow_Control_Enable command succeeded
0x01 to 0xFF	HCI_Read_Synchronous_Flow_Control_Enable command failed (see [Vol 1] Part F, Controller Error Codes for list of error codes).

Synchronous_Flow_Control_Enable:

Size: 1 octet

Value	Parameter Description
0x00	Synchronous Flow Control is disabled. No HCI_Number_Of_Completed_-Packets events will be sent from the Controller for synchronous Connection_Handles.
0x01	Synchronous Flow Control is enabled. HCI_Number_Of_Completed_-Packets events will be sent from the Controller for synchronous Connection_Handles.

Event(s) generated (unless masked away):

When the HCI_Read_Synchronous_Flow_Control_Enable command has completed an HCI_Command_Complete event shall be generated.



7.3.37 Write Synchronous Flow Control Enable command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_- Synchronous_Flow_- Control_Enable	0x002F	Synchronous_Flow_Con- trol_Enable	Status

Description:

The HCI_Write_Synchronous_Flow_Control_Enable command provides the ability to write the Synchronous_Flow_Control_Enable parameter. See [Section 6.22](#).

The Synchronous_Flow_Control_Enable parameter can only be changed if no connections exist.

Command parameters:

Synchronous_Flow_Control_Enable:

Size: 1 octet

Value	Parameter Description
0x00	Synchronous Flow Control is disabled. No HCI_Number_Of_Completed_Packets events shall be sent from the Controller for synchronous Connection_Handles. Default
0x01	Synchronous Flow Control is enabled. HCI_Number_Of_Completed_Packets events shall be sent from the Controller for synchronous Connection_Handles.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Write_Synchronous_Flow_Control_Enable command succeeded
0x01 to 0xFF	HCI_Write_Synchronous_Flow_Control_Enable command failed (see [Vol 1] Part F, Controller Error Codes for list of error codes.)

Event(s) generated (unless masked away):

When the HCI_Write_Synchronous_Flow_Control_Enable command has completed an HCI_Command_Complete event shall be generated.



7.3.38 Set Controller To Host Flow Control command

Command	OCF	Command Parameters	Return Parameters
HCI_Set_Controller_To_Host_Flow_Control	0x0031	Flow_Control_Enable	Status

Description:

This command is used by the Host to turn flow control on or off for data and/or voice sent in the direction from the Controller to the Host. If flow control is turned off, the Host should not send the HCI_Host_Number_Of_Completed_Packets command. That command will be ignored by the Controller if it is sent by the Host and flow control is off. If flow control is turned on for HCI ACL Data packets and off for HCI Synchronous Data packets, HCI_Host_Number_Of_Completed_Packets commands sent by the Host should only contain Connection_Handles for ACL connections. If flow control is turned off for HCI ACL Data packets and on for HCI Synchronous Data packets, HCI_Host_Number_Of_Completed_Packets commands sent by the Host should only contain Connection_Handles for synchronous connections. If flow control is turned on for HCI ACL Data packets and HCI Synchronous Data packets, the Host will send HCI_Host_Number_Of_Completed_Packets commands both for ACL connections and synchronous connections.

The Flow_Control_Enable parameter shall only be changed if no connections exist.

Command parameters:

Flow_Control_Enable:

Size: 1 octet

Value	Parameter Description
0x00	Flow control off in direction from Controller to Host. Default.
0x01	Flow control on for HCI ACL Data packets and off for HCI Synchronous Data packets in direction from Controller to Host.
0x02	Flow control off for HCI ACL Data packets and on for HCI Synchronous Data packets in direction from Controller to Host.
0x03	Flow control on both for HCI ACL Data packets and HCI Synchronous Data packets in direction from Controller to Host.
All other values	Reserved for future use

**Return parameters:***Status:**Size: 1 octet*

Value	Parameter Description
0x00	HCI_Set_Controller_To_Host_Flow_Control command succeeded.
0x01 to 0xFF	HCI_Set_Controller_To_Host_Flow_Control command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

When the HCI_Set_Controller_To_Host_Flow_Control command has completed, an HCI_Command_Complete event shall be generated.



7.3.39 Host Buffer Size command

Command	OCF	Command Parameters	Return Parameters
HCI_Host_Buffer_Size	0x0033	Host_ACL_Data_Packet_Length, Host_Synchronous_Data_Packet_Length, Host_Total_Num_ACL_Data_Packets, Host_Total_Num_Synchronous_Data_Packets	Status

Description:

The HCI_Host_Buffer_Size 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 to the Host. The Controller shall segment the data to be transmitted from the Controller to the Host according to these sizes, so that the HCI Data packets will contain data with up to these sizes. The HCI_Host_Buffer_Size command also notifies the Controller about the total number of HCI ACL and Synchronous Data packets that can be stored in the data buffers of the Host. If flow control from the Controller to the Host is turned off, and the HCI_Host_Buffer_Size command has not been issued by the Host, this means that the Controller will send HCI Data packets to the Host with any lengths the Controller wants to use, and it is assumed that the data buffer sizes of the Host are unlimited. If flow control from the Controller to the Host is turned on, the HCI_Host_Buffer_Size command shall after a power-on or a reset always be sent by the Host before the first HCI_Host_Number_Of_Completed_Packets command is sent.

The [Set Controller To Host Flow Control command](#) is used to turn flow control on or off. The Host_ACL_Data_Packet_Length command parameter will be 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 command parameter is used to determine the maximum size of HCI Synchronous Data packets. Both the Host and the Controller shall 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 command parameter contains the total number of HCI ACL Data packets that can be stored in the data buffers of the Host. The Controller will determine how the buffers are to be divided between different Connection_Handles. The Host_Total_Num_Synchronous_Data_Packets command parameter gives the same information for HCI Synchronous Data packets.

Note: The Host_ACL_Data_Packet_Length and Host_Synchronous_Data_Packet_Length command parameters do not include the length of the HCI ACL Data packet header or the HCI Synchronous Data packet header respectively.

**Command parameters:***Host_ACL_Data_Packet_Length:**Size: 2 octets*

Value	Parameter Description
0xFFFF	Maximum length (in octets) of the data portion of each HCI ACL Data packet that the Host is able to accept.

*Host_Synchronous_Data_Packet_Length:**Size: 1 octet*

Value	Parameter Description
0xFF	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:**Size: 2 octets*

Value	Parameter Description
0xFFFF	Total number of HCI ACL Data packets that can be stored in the data buffers of the Host.

*Host_Total_Num_Synchronous_Data_Packets:**Size: 2 octets*

Value	Parameter Description
0xFFFF	Total number of HCI Synchronous Data packets that can be stored in the data buffers of the Host.

Return parameters:*Status:**Size: 1 octet*

Value	Parameter Description
0x00	HCI_Host_Buffer_Size command succeeded.
0x01 to 0xFF	HCI_Host_Buffer_Size command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

When the HCI_Host_Buffer_Size command has completed, an HCI_Command_Complete event shall be generated.



7.3.40 Host Number Of Completed Packets command

Command	OCF	Command Parameters	Return Parameters
HCI_Host_Number_Of_Completed_Packets	0x0035	Num_Handles, Connection_Handle[i], Host_Num_Completed_Packets[i]	

Description:

The HCI_Host_Number_Of_Completed_Packets command is used by the Host to indicate to the Controller the number of HCI Data packets that have been 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 in the Host and is available for new packets to be sent. Based on this information, and the Host_Total_Num_ACL_Data_Packets and Host_Total_Num_Synchronous_Data_Packets command parameters of the HCI_Host_Buffer_Size command, the Controller can determine for which Connection_Handles the following HCI Data packets should be sent to the Host. When the Host has completed one or more HCI Data packet(s) it shall send an 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 [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 shall always be sent by the Host after a power-on or a reset before the first HCI_Host_Number_Of_Completed_Packets command is sent.

The HCI_Host_Number_Of_Completed_Packets command may be sent at any time by the Host when there is at least one connection, or if the Controller is in local loopback mode, independent of other commands. If the Host issues this command when neither of these cases applies, the Controller shall ignore it.

**Command parameters:***Num_Handles:**Size: 1 octet*

Value	Parameter Description
0xXX	The number of Connection_Handles and Host_Num_Completed_Packets parameters pairs contained in this command. Range: 0 to 255

Connection_Handle[i]: *Size: Num_Handles × 2 octets (12 bits meaningful)*

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

Host_Num_Completed_Packets[i]: *Size: Num_Handles × 2 octets*

Value	Parameter Description
0xFFFF	The number of HCI Data packets that have been completed for the associated Connection_Handle since the previous time the event was returned. Range: 0x0000 to 0xFFFF

Return parameters:

None.

Event(s) generated (unless masked away):

Normally, no event is generated after the HCI_Host_Number_Of_Completed_Packets command has completed. However, if the HCI_Host_Number_Of_Completed_Packets command contains one or more invalid parameters, the Controller shall return an HCI_Command_Complete event containing the error code *Invalid HCI Command Parameters* (0x12). The normal flow control for commands is not used for this command.



7.3.41 Read Link Supervision Timeout command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Link_Supervision_Timeout	0x0036	Handle	Status, Handle, Link_Supervision_Timeout

Description:

This command reads the value for the Link_Supervision_Timeout parameter for the Controller.

The Handle used for this command shall be the ACL connection to the appropriate device. See [Section 6.21](#).

Command parameters:

Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xFFFF	Specifies which Connection_Handle's Link Supervision Timeout value is to be read. The Handle is a Connection_Handle for a BR/EDR Controller. Range: 0x0000 to 0x0EFF

Return parameters:

Status: *Size: 1 octet*

Value	Parameter Description
0x00	HCI_Read_Link_Supervision_Timeout command succeeded.
0x01 to 0xFF	HCI_Read_Link_Supervision_Timeout command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xFFFF	Specifies which Connection_Handle's Link Supervision Timeout value was read. The Handle is a Connection_Handle for a BR/EDR Controller. Range: 0x0000 to 0x0EFF

**Link_Supervision_Timeout:****Size: 2 octets**

Value	Parameter Description
0x0000	No Link_Supervision_Timeout.
N = 0xXXXX	Measured in number of BR/EDR Baseband slots Link_Supervision_Timeout = N * 0.625 ms (1 Baseband slot) Range: 0x0001 to 0xFFFF Time Range: 0.625 ms to 40.9 s

Event(s) generated (unless masked away):

When the HCI_Read_Link_Supervision_Timeout command has completed, an HCI_Command_Complete event shall be generated.



7.3.42 Write Link Supervision Timeout command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Link_Supervision_Timeout	0x0037	Handle, Link_Supervision_Timeout	Status, Handle

Description:

This command writes the value for the Link_Supervision_Timeout parameter for a BR/EDR Controller. If the Host issues this command with a Connection_Handle where the Controller is the Peripheral, the Controller shall return the error code *Command Disallowed* (0x0C).

The Handle used for this command shall be the ACL connection to the appropriate device. This command will set the Link_Supervision_Timeout values for other Synchronous Handles to that device. See [Section 6.21](#).

Command parameters:

Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xFFFF	Specifies which Handle's Link Supervision Timeout value is to be written. The Handle is a Connection_Handle for a BR/EDR Controller. Range: 0x0000 to 0xFFFF

Link_Supervision_Timeout: *Size: 2 octets*

Value	Parameter Description
0x0000	No Link_Supervision_Timeout.
N = 0xFFFF	Measured in number of BR/EDR Baseband slots Link_Supervision_Timeout = N*0.625 ms (1 Baseband slot) Range: 0x0001 to 0xFFFF Time Range: 0.625 ms to 40.9 s Default: 0x7D00 Link_Supervision_Timeout = 20 s Mandatory Range for Controller: 0x0190 to 0xFFFF; plus 0 for infinite timeout

**Return parameters:***Status:**Size: 1 octet*

Value	Parameter Description
0x00	HCI_Write_Link_Supervision_Timeout command succeeded.
0x01 to 0xFF	HCI_Write_Link_Supervision_Timeout command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

*Handle:**Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xFFFF	Specifies which Handle's Link Supervision Timeout value was written. The Handle is a Connection_Handle for a BR/EDR Controller. Range: 0x0000 to 0x0EFF

Event(s) generated (unless masked away):

When the HCI_Write_Link_Supervision_Timeout command has completed, an HCI_Command_Complete event shall be generated.



7.3.43 Read Number Of Supported IAC command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Number_Of_Supported_IAC	0x0038		Status, Num_Support- ed_IAC

Description:

This command reads the value for the number of Inquiry Access Codes (IAC) that the local BR/EDR Controller can simultaneous listen for during an Inquiry Scan. All BR/EDR Controllers are required to support at least one IAC, the General Inquiry Access Code (the GIAC). Some BR/EDR Controllers support additional IACs.

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Number_Of_Supported_IAC command succeeded.
0x01 to 0xFF	HCI_Read_Number_Of_Supported_IAC command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Num_Supported_IAC:

Size: 1 octet

Value	Parameter Description
0xXX	Specifies the number of Supported IAC that the local BR/EDR Controller can simultaneous listen for during an Inquiry Scan. Range: 0x01 to 0x40

Event(s) generated (unless masked away):

When the HCI_Read_Number_Of_Supported_IAC command has completed, an HCI_Command_Complete event shall be generated.



7.3.44 Read Current IAC LAP command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Current_IAC_LAP	0x0039		Status, Num_Current_IAC, IAC_LAP[i]

Description:

This command reads the LAP(s) used to create the Inquiry Access Codes (IAC) that the local BR/EDR Controller is simultaneously scanning for during Inquiry Scans. All BR/EDR Controllers shall support at least one IAC, the General Inquiry Access Code (the GIAC). Some BR/EDR Controllers support additional IACs.

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Current_IAC_LAP command succeeded.
0x01 to 0xFF	HCI_Read_Current_IAC_LAP command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Num_Current_IAC:

Size: 1 octet

Value	Parameter Description
0xXX	Specifies the number of IACs which are currently in use by the local BR/EDR Controller to simultaneously listen for during an Inquiry Scan. Range: 0x01 to 0x40

IAC_LAP[i]:

Size: Num_Current_IAC × 3 octets

Value	Parameter Description
0XXXXXXXX	LAP used to create the IAC which is currently in use by the local BR/EDR Controller to simultaneously listen for during an Inquiry Scan. Range: 0x9E8B00 to 0x9E8B3F

Event(s) generated (unless masked away):

When the HCI_Read_Current_IAC_LAP command has completed, an HCI_Command_Complete event shall be generated.



7.3.45 Write Current IAC LAP command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Current_IAC_LAP	0x003A	Num_Current_IAC, IAC_LAP[i]	Status

Description:

This command writes the LAP(s) used to create the Inquiry Access Codes (IAC) that the local BR/EDR Controller is simultaneously scanning for during Inquiry Scans. All BR/EDR Controller shall support at least one IAC, the General Inquiry Access Code (the GIAC). Some BR/EDR Controllers support additional IACs.

This command shall clear any existing IACs and stores Num_Current_IAC and the IAC_LAPs in to the Controller. If Num_Current_IAC is greater than Num_Supported_IAC then only the first Num_Supported_IAC shall be stored in the Controller, and an HCI_Command_Complete event with error code *Success* (0x00) shall be generated.

Command parameters:

Num_Current_IAC:

Size: 1 octet

Value	Parameter Description
0xXX	Specifies the number of IACs that will be used by the local BR/EDR Controller to simultaneously listen for during an Inquiry Scan. Range: 0x01 to 0x40

IAC_LAP[i]:

Size: Num_Current_IAC × 3 octets

Value	Parameter Description
0XXXXXXXX	LAP that will be used to create the IACs that will be used by the local BR/EDR Controller to simultaneously listen for during an Inquiry Scan. Range: 0x9E8B00 to 0x9E8B3F. The default IAC(s) to be used shall be the GIAC and zero or more other IACs specified by the manufacturer.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Write_Current_IAC_LAP command succeeded.
0x01 to 0xFF	HCI_Write_Current_IAC_LAP command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

**Event(s) generated (unless masked away):**

When the HCI_Write_Current_IAC_LAP command has completed, an HCI_Command_Complete event shall be generated.



7.3.46 Set AFH Host Channel Classification command

Command	OCF	Command Parameters	Return Parameters
HCI_Set_AFH_Host_Channel_- Classification	0x003F	AFH_Host_Channel_- Classification	Status

Description:

The `HCI_Set_AFH_Host_Channel_Classification` command allows the Host to specify a channel classification based on its “local information”. This classification persists until overwritten with a subsequent `HCI_Set_AFH_Host_Channel_Classification` command or until the BR/EDR Controller is reset.

If this command is used, updates should be sent within 10 seconds of the Host knowing that the channel classification has changed. The interval between two successive commands sent shall be at least 1 second.

Command parameters:

AFH_Host_Channel_Classification: *Size: 10 octets (79 bits meaningful)*

Value	Parameter Description
0xFFFFFFFF XXXXXXXXXX XX	<p>This parameter contains 80 1-bit fields.</p> <p>The n^{th} such field (in the range 0 to 78) contains the value for channel n:</p> <p>0: channel n is bad</p> <p>1: channel n is unknown</p> <p>The most significant bit (bit 79) is reserved for future use</p> <p>At least N_{\min} channels shall be marked as unknown. (See [Vol 2] Part B, Section 2.3.1). If the device supports Synchronizable mode, then the synchronization train channels (see [Vol 2] Part B, Section 2.6.4.8) shall be excluded when checking this requirement.</p>

Return parameters:

Status: *Size: 1 octet*

Value	Parameter Description
0x00	HCI_Set_AFH_Host_Channel_Classification command succeeded.
0x01 to 0xFF	HCI_Set_AFH_Host_Channel_Classification command failed. [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

When the HCI_Set_AFH_Host_Channel_Classification command has completed, an HCI Command Complete event shall be generated.



7.3.47 Read Inquiry Scan Type command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Inquiry_Scan_Type	0x0042		Status, Inquiry_Scan_Type

Description:

This command reads the Inquiry_Scan_Type configuration parameter from the local BR/EDR Controller. See [Section 6.4](#).

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Inquiry_Scan_Type command succeeded
0x01 to 0xFF	HCI_Read_Inquiry_Scan_Type command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Inquiry_Scan_Type:

Size: 1 octet

Value	Parameter Description
0x00	Standard Scan (default)
0x01	Interlaced Scan
All other values	Reserved for future use

Event(s) generated (unless masked away):

When the HCI_Read_Inquiry_Scan_Type command has completed, an HCI_Command_Complete event shall be generated.



7.3.48 Write Inquiry Scan Type command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Inquiry_Scan_Type	0x0043	Scan_Type	Status

Description:

This command writes the Inquiry Scan Type configuration parameter of the local BR/EDR Controller. See [Section 6.4](#).

Command parameters:

Scan_Type: *Size: 1 octet*

Value	Parameter Description
0x00	Standard Scan (default)
0x01	Interlaced Scan
All other values	Reserved for future use

Return parameters:

Status: *Size: 1 octet*

Value	Parameter Description
0x00	HCI_Write_Inquiry_Scan_Type command succeeded
0x01 to 0xFF	HCI_Write_Inquiry_Scan_Type command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

When the HCI_Write_Inquiry_Scan_Type command has completed, an HCI_Command_Complete event shall be generated.



7.3.49 Read Inquiry Mode command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Inquiry_Mode	0x0044		Status, Inquiry_ Mode

Description:

This command reads the Inquiry_Mode configuration parameter of the local BR/EDR Controller. See [Section 6.5](#).

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Inquiry_Mode command succeeded.
0x01 to 0xFF	HCI_Read_Inquiry_Mode command failed. See [Vol 1] Part F, Controller Error Codes for list of error codes.

Inquiry_Mode:

Size: 1 octet

Value	Parameter Description
0x00	Standard Inquiry Result event format
0x01	Inquiry Result format with RSSI
0x02	Inquiry Result with RSSI format or Extended Inquiry Result format
All other values	Reserved for future use

Event(s) generated (unless masked away):

When the HCI_Read_Inquiry_Mode command has completed, an HCI_Command_Complete event shall be generated.



7.3.50 Write Inquiry Mode command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Inquiry_Mode	0x0045	Inquiry_Mode	Status

Description:

This command writes the Inquiry_Mode configuration parameter of the local BR/EDR Controller. See [Section 6.5](#).

Command parameters:

Inquiry_Mode:

Size: 1 octet

Value	Parameter Description
0x00	Standard Inquiry Result event format (default)
0x01	Inquiry Result format with RSSI
0x02	Inquiry Result with RSSI format or Extended Inquiry Result format
All other values	Reserved for future use

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Write_Inquiry_Mode command succeeded.
0x01 to 0xFF	HCI_Write_Inquiry_Mode command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

When the HCI_Write_Inquiry_Mode command has completed, an HCI_Command_Complete event shall be generated.



7.3.51 Read Page Scan Type command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Page_Scan_Type	0x0046		Status, Page_Scan_Type

Description:

This command reads the Page Scan Type configuration parameter of the local BR/EDR Controller. See [Section 6.11](#).

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Page_Scan_Type command succeeded.
0x01 to 0xFF	HCI_Read_Page_Scan_Type command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Page_Scan_Type:

Size: 1 octet

Value	Parameter Description
0x00	Mandatory: Standard Scan (default)
0x01	Optional: Interlaced Scan
All other values	Reserved for future use

Event(s) generated (unless masked away):

When the HCI_Read_Page_Scan_Type command has completed, an HCI_Command_Complete event shall be generated.



7.3.52 Write Page Scan Type command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Page_Scan_Type	0x0047	Page_Scan_Type	Status

Description:

This command writes the Page Scan Type configuration parameter of the local BR/EDR Controller. See [Section 6.11](#).

Command parameters:

Page_Scan_Type:

Size: 1 octet

Value	Parameter Description
0x00	Mandatory: Standard Scan (default)
0x01	Optional: Interlaced Scan
All other values	Reserved for future use

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Write_Page_Scan_Type command succeeded.
0x01 to 0xFF	HCI_Write_Page_Scan_Type command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

When the HCI_Write_Page_Scan_Type command has completed, an HCI_Command_Complete event shall be generated.



7.3.53 Read AFH Channel Assessment Mode command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_AFH_Channel_Assessment_Mode	0x0048		Status, AFH_Channel_Assessment_Mode

Description:

The HCI_Read_AFH_Channel_Assessment_Mode command reads the value for the AFH_Channel_Assessment_Mode parameter. The AFH_Channel_Assessment_Mode parameter controls whether the Controller's channel assessment scheme is enabled or disabled.

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_AFH_Channel_Assessment_Mode command succeeded.
0x01 to 0xFF	HCI_Read_AFH_Channel_Assessment_Mode command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

AFH_Channel_Assessment_Mode:

Size: 1 octet

Value	Parameter Description
0x00	Controller channel assessment disabled.
0x01	Controller channel assessment enabled.
All other values	Reserved for future use.

Event(s) generated (unless masked away):

When the HCI_Read_AFH_Channel_Assessment_Mode command has completed, an HCI_Command_Complete event shall be generated.



7.3.54 Write AFH Channel Assessment Mode command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_AFH_Channel_Assessment_Mode	0x0049	AFH_Channel_Assessment_Mode	Status

Description:

The HCI_Write_AFH_Channel_Assessment_Mode command writes the value for the AFH_Channel_Assessment_Mode parameter. The AFH_Channel_Assessment_Mode parameter controls whether the Controller's channel assessment scheme is enabled or disabled.

Disabling channel assessment forces all channels to be unknown in the local classification for the BR/EDR physical transport, but does not affect the AFH_reporting_mode or support for the HCI_Set_AFH_Host_Channel_Classification command. A BR/EDR Peripheral in the AFH_reporting_enabled state shall continue to send LMP channel classification messages for any changes to the channel classification caused by either this command (altering the AFH_Channel_Assessment_Mode) or HCI_Set_AFH_Host_Channel_Classification command (providing a new channel classification from the Host).

Disabling channel assessment also forces all channels to be unknown in the local classification for the LE physical transport. If channel classification reporting is enabled by the Central, then the following rules apply to the Peripheral:

- Irrespective of whether channel assessment is enabled or disabled by the Host, the Controller shall continue to send LL_CHANNEL_STATUS_IND PDUs for any changes to the channel classification caused by the HCI_LE_Set_Host_Channel_Classification command.
- If channel assessment has been enabled by the Host, the Controller shall send LL_CHANNEL_STATUS_IND PDUs for any changes to the channel classification caused by the HCI_LE_Set_Host_Channel_Classification command and for any changes reported by the channel assessment scheme.
- The Controller shall send an LL_CHANNEL_STATUS_IND PDU whenever the channel classification changes because this command changes the channel assessment mode.

If the AFH_Channel_Assessment_Mode parameter is enabled and the Controller does not support a channel assessment scheme, other than via the HCI_Set_AFH_Host_Channel_Classification command (for BR/EDR) or via the HCI_LE_Set_Host_Channel_Classification command (for LE), then a Status parameter of 'Channel Assessment Not Supported' should be returned. See [Part F, Controller Error Codes](#) for a list of error codes and descriptions.



If the Controller supports a channel assessment scheme then the default AFH_Channel_Assessment_Mode is enabled, otherwise the default is disabled.

Command parameters:*AFH_Channel_Assessment_Mode:**Size: 1 octet*

Value	Parameter Description
0x00	Controller channel assessment disabled.
0x01	Controller channel assessment enabled.
All other values	Reserved for future use.

Return parameters:*Status:**Size: 1 octet*

Value	Parameter Description
0x00	HCI_Write_AFH_Channel_Assessment_Mode command succeeded.
0x01 to 0xFF	HCI_Write_AFH_Channel_Assessment_Mode command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

When the HCI_Write_AFH_Channel_Assessment_Mode command has completed, an HCI_Command_Complete event shall be generated.



7.3.55 Read Extended Inquiry Response command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Extended_Inquiry_Response	0x0051		Status, FEC_Required, Extended_Inquiry_Response

Description:

The HCI_Read_Extended_Inquiry_Response command reads the extended inquiry response to be sent during the extended inquiry response procedure. The FEC_Required parameter states if FEC encoding is required.

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Extended_Inquiry_Response command succeeded.
0x01 to 0xFF	HCI_Read_Extended_Inquiry_Response command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

FEC_Required:

Size: 1 octet

Value	Parameter Description
0x00	FEC is not required
0x01	FEC is required
All other values	Reserved for future use

Extended_Inquiry_Response:

Size: 240 octets

Value	Parameter Description
	Extended inquiry response data as defined in [Vol 3] Part C, Section 8 .

Event(s) generated (unless masked away):

When the HCI_Read_Extended_Inquiry_Response command has completed, an HCI_Command_Complete event shall be generated.



7.3.56 Write Extended Inquiry Response command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Extended_Inquiry_Response	0x0052	FEC_Required, Extended_Inquiry_Response	Status

Description:

The HCI_Write_Extended_Inquiry_Response command writes the extended inquiry response to be sent during the extended inquiry response procedure. The FEC_Required command parameter states if FEC encoding is required. The extended inquiry response data is not preserved over a reset. The initial value of the inquiry response data is all zero octets. The Controller shall not interpret the extended inquiry response data.

Command parameters:

FEC_Required:

Size: 1 octet

Value	Parameter Description
0x00	FEC is not required
0x01	FEC is required
All other values	Reserved for future use

Extended_Inquiry_Response:

Size: 240 octets

Value	Parameter Description
	Extended inquiry response data as defined in [Vol 3] Part C, Section 8 .
	All octets zero (default).

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Write_Extended_Inquiry_Response command succeeded
0x01 to 0xFF	HCI_Write_Extended_Inquiry_Response command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

When the HCI_Write_Extended_Inquiry_Response command has completed, an HCI_Command_Complete event shall be generated.



7.3.57 Refresh Encryption Key command

Command	OCF	Command Parameters	Return Parameters
HCI_Refresh_Encryption_Key	0x0053	Connection_Handle	

Description:

This command is used by the Host to cause the BR/EDR Controller to refresh the encryption key on an ACL connection identified by a Connection_Handle by pausing and resuming encryption.

Command parameters:

Connection_Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xXXXX	Connection_Handle Range: 0x0000 to 0x0EFF

Return parameters:

None.

Event(s) generated (unless masked away):

An HCI_Command_Status event is sent from the BR/EDR Controller to the Host when the Controller has started the Refresh Encryption Key procedure. An HCI_Encryption_Key_Refresh_Complete event shall be generated when the Refresh Encryption Key procedure has completed.



7.3.58 Read Simple Pairing Mode command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Simple_Pairing_Mode	0x0055		Status, Simple_Pairing_Mode

Description:

This command reads the Simple_Pairing_Mode parameter in the BR/EDR Controller.

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Simple_Pairing_Mode command succeeded.
0x01 to 0xFF	HCI_Read_Simple_Pairing_Mode command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Simple_Pairing_Mode:

Size: 1 octet

Value	Parameter Description
0x00	Secure Simple Pairing not set (default)
0x01	Secure Simple Pairing enabled
All other values	Reserved for future use

Event(s) generated (unless masked away):

When the HCI_Read_Simple_Pairing_Mode command has completed, an HCI_Command_Complete event shall be generated.



7.3.59 Write Simple Pairing Mode command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Simple_Pairing_Mode	0x0056	Simple_Pairing_Mode	Status

Description:

This command enables Secure Simple Pairing mode in the BR/EDR Controller. When Secure Simple Pairing Mode is set to 'enabled' the Link Manager shall respond to an LMP_IO_CAPABILITY_REQ PDU with an LMP_IO_CAPABILITY_RES PDU and continue with the subsequent pairing procedure. When Secure Simple Pairing mode is set to 'disabled', the Link Manager shall reject an IO capability request. A Host shall not set the Secure Simple Pairing Mode to 'disabled.'

Until Write_Simple_Pairing_Mode is received by the BR/EDR Controller, it shall not support any Secure Simple Pairing sequences, and shall return the error code *Secure Simple Pairing not Supported by Host* (0x37). This command shall be written before initiating page scan or paging procedures.

The Link Manager Secure Simple Pairing (Host Support) feature bit shall be set to the Simple_Pairing_Mode parameter. The default value for Simple_Pairing_Mode shall be 'disabled.' When Simple_Pairing_Mode is set to 'enabled,' the bit in the LMP features mask indicating support for Secure Simple Pairing (Host Support) shall be set to enabled in subsequent responses to an LMP_FEATURES_REQ from a remote device.

Command parameters:

Simple_Pairing_Mode:

Size: 1 octet

Value	Parameter Description
0x00	Secure Simple Pairing disabled (default)
0x01	Secure Simple Pairing enabled
All other values	Reserved for future use

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Write_Simple_Pairing_Mode command succeeded.
0x01 to 0xFF	HCI_Write_Simple_Pairing_Mode command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

**Event(s) generated (unless masked away):**

When the HCI_Write_Simple_Pairing_Mode command has completed, an HCI_Command_Complete event shall be generated.



7.3.60 Read Local OOB Data command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Local_OOB_Data	0x0057		Status, C, R

Description:

This command obtains a Secure Simple Pairing Hash C and Randomizer R which are intended to be transferred to a remote device using an OOB mechanism. The BR/EDR Controller shall create new values for C and R for each invocation of this command.

Note: Each OOB transfer will have unique C and R values.

After each OOB transfer this command shall be used to obtain a new set of values for the next OOB transfer.

Note: The Controller keeps information used to generate these values for later use in the Secure Simple Pairing process. If the BR/EDR Controller is powered off or reset then this information is lost and the values obtained before the power off or reset are invalid.

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Local_OOB_Data command succeeded.
0x01 to 0xFF	HCI_Read_Local_OOB_Data command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

C:

Size: 16 octets

Value	Parameter Description
0xFFFFFFFFXX XXXXXXXXXXXX XXXXXXXXXXXX	Secure Simple Pairing Hash C

**R:****Size: 16 octets**

Value	Parameter Description
0xFFFFFFFFXXXXXXX XXXXXXXXXXXX XXXXXXXXXXXX	Secure Simple Pairing Randomizer R

Event(s) generated (unless masked away):

When the HCI_Read_Local_OOB_Data command has completed, an HCI_Command_Complete event shall be generated.



7.3.61 Read Inquiry Response Transmit Power Level command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Inquiry_Response_Transmit_Power_Level	0x0058		Status, TX_Power

Description:

This command reads the power level used to transmit the FHS and EIR data packets. This can be used directly in the Tx Power Level EIR data type.

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Inquiry_Response_Transmit_Power_Level command succeeded.
0x01 to 0xFF	HCI_Read_Inquiry_Response_Transmit_Power_Level command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

TX_Power:

Size: 1 octet

Value	Parameter Description
0xXX	Range: -70 to 20 Units: dBm

Event(s) generated (unless masked away):

When the HCI_Read_Inquiry_Response_Transmit_Power_Level command has completed, an HCI_Command_Complete event shall be generated.



7.3.62 Write Inquiry Transmit Power Level command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Inquiry_Transmit_Power_Level	0x0059	TX_Power	Status

Description:

This command writes the inquiry transmit power level used to transmit the inquiry (ID) data packets. The Controller should use the supported TX power level closest to the TX_Power parameter.

Command parameters:

TX_Power:

Size: 1 octet

Value	Parameter Description
0xXX	Range: -70 to 20 Units: dBm

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Write_Inquiry_Transmit_Power_Level command succeeded
0x01 to 0xFF	HCI_Write_Inquiry_Transmit_Power_Level command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

When the HCI_Write_Inquiry_Transmit_Power_Level command has completed, an HCI_Command_Complete event shall be generated.



7.3.63 Send Keypress Notification command

Command	OCF	Command Parameters	Return Parameters
HCI_Send_Keypress_Notification	0x0060	BD_ADDR, Notification_Type	Status, BD_ADDR

Description:

This command is used during the Passkey Entry protocol by a device with KeyboardOnly IO capabilities. It is used by a Host to inform the remote device when keys have been entered or erased.

Command parameters:

BD_ADDR: *Size: 6 octets*

Value	Parameter Description
0xFFFFFFFFXXXX	BD_ADDR of remote device involved in the Secure Simple Pairing process

Notification_Type: *Size: 1 octet*

Value	Parameter Description
0	Passkey entry started
1	Passkey digit entered
2	Passkey digit erased
3	Passkey cleared
4	Passkey entry completed
All other values	Reserved for future use

Return parameters:

Status: *Size: 1 octet*

Value	Parameter Description
0x00	HCI_Send_Keypress_Notification command succeeded
0x01 to 0xFF	HCI_Send_Keypress_Notification command failed

BD_ADDR: *Size: 6 octets*

Value	Parameter Description
0xFFFFFFFFXXXX	BD_ADDR of remote device involved in the Secure Simple Pairing process

**Event(s) generated (unless masked away):**

When the HCI_Send_Keypress_Notification command has completed, an HCI_Command_Complete event shall be generated.



7.3.64 Read Default Erroneous Data Reporting command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Default_-_Erroneous_Data_Reporting	0x005A		Status, Erroneous_Data_Reporting

Description:

This command reads the Erroneous_Data_Reporting parameter.

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Default_Erroneous_Data_Reporting command succeeded.
0x01 to 0xFF	HCI_Read_Default_Erroneous_Data_Reporting command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Erroneous_Data_Reporting:

Size: 1 octet

Value	Parameter Description
0x00	Erroneous data reporting disabled.
0x01	Erroneous data reporting enabled.
All other values	Reserved for future use.

Event(s) generated (unless masked away):

When the HCI_Read_Default_Erroneous_Data_Reporting command has completed, an HCI_Command_Complete event shall be generated.



7.3.65 Write Default Erroneous Data Reporting command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Default_-_Erroneous_Data_Reporting	0x005B	Erroneous_Data_Reporting	Status

Description:

This command writes the Erroneous_Data_Reporting parameter. The BR/EDR Controller shall set the Packet_Status_Flag as defined in [Section 5.4.3 HCI Synchronous Data packets](#), depending on the value of this parameter. The new value for the Erroneous_Data_Reporting parameter shall not apply to existing synchronous connections.

Command parameters:

Erroneous_Data_Reporting:

Size: 1 octet

Value	Parameter Description
0x00	Erroneous Data reporting disabled (default).
0x01	Erroneous data reporting enabled.
All other values	Reserved for future use.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Write_Default_Erroneous_Data_Reporting command succeeded.
0x01 to 0xFF	HCI_Write_Default_Erroneous_Data_Reporting command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

When the HCI_Write_Default_Erroneous_Data_Reporting command has completed, an HCI_Command_Complete event shall be generated.



7.3.66 Enhanced Flush command

Command	OCF	Command Parameters	Return Parameters
HCI_Enhanced_Flush	0x005F	Handle, Packet_Type	

Description:

The HCI_Enhanced_Flush command is used to discard all L2CAP packets identified by Packet_Type that are currently pending for transmission in the Controller for the specified Handle, even if there currently are chunks of data that belong to more than one L2CAP packet of the same type in the Controller. The only packet type defined is automatically-flushable. Packets not identified by Packet_Type will not be flushed and will be processed normally by the Controller.

After flushing the packets, all data that is sent to the BR/EDR Controller for the same Handle and packet type shall be discarded by the Controller until an HCI ACL Data packet with the start Packet_Boundary_Flag (0x00 or 0x02) is received. This command allows higher-level software to control how long the Baseband should try to retransmit a Baseband packet of a specific type for a Handle before all data of that type currently pending for transmission in the Controller should be flushed. The HCI_Enhanced_Flush command shall be used for ACL-U connections only. On the BR/EDR Controller, the HCI_Flush command can be used to flush all packets (see [Section 7.3.4](#)). In addition to the HCI_Enhanced_Flush and HCI_Flush commands, the automatic flush timers (see [Section 7.3.29](#)) can be used to automatically flush an automatically-flushable L2CAP packet that is currently being transmitted after the specified flush timer has expired.

Command parameters:

Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xXXXX	Handle to be used to identify a connection. Range: 0x0000 to 0x0EFF

**Packet_Type:****Size: 1 octet**

Value	Parameter Description
0x00	Automatically flushable only.
All other values	Reserved for future use.

Return parameters:

None.

Event(s) generated (unless masked away):

When the Controller receives the HCI_Enhanced_Flush command, the Controller shall send the HCI_Command_Status event to the Host. In addition, when all the packets identified by Packet_Type have been flushed for the specified Handle, the Controller shall send an HCI_Enhanced_Flush_Complete event to the Host. The Controller may send the HCI_Enhanced_Flush_Complete event immediately after flushing all the packets of type Packet_Type for the specified Handle, or it may wait until all packets for the specified Handle, independent of Packet_Type, buffered in the Controller at the time of the receipt of the HCI_Enhanced_Flush command, have been either flushed or transmitted.



7.3.67 [This section is no longer used]

7.3.68 [This section is no longer used]



7.3.69 Set Event Mask Page 2 command

Command	OCF	Command Parameters	Return Parameters
HCI_Set_Event_Mask_Page_2	0x0063	Event_Mask_Page_2	Status

Description:

The HCI_Set_Event_Mask_Page_2 command is used to control which events are generated by the HCI for the Host. The Event_Mask_Page_2 is a logical extension to the Event_Mask parameter of the HCI_Set_Event_Mask command. If the bit in the Event_Mask_Page_2 is set to a one, then the event associated with that bit shall be enabled. The event mask allows the Host to control how much it is interrupted.

The Controller shall ignore those bits which are reserved for future use or represent events which it does not support. If the Host sets any of these bits to 1, the Controller shall act as if they were set to 0.

Command parameters:

Event_Mask_Page_2:

Size: 8 octets

Bit	Parameter Description
0	Previously used
1	Previously used
2	Previously used
3	Previously used
4	Previously used
5	Previously used
6	Previously used
7	Previously used
8	Number Of Completed Data Blocks event
9	Previously used
10	Previously used
11	Previously used
12	Previously used
13	Previously used
14	Triggered Clock Capture event
15	Synchronization Train Complete event
16	Synchronization Train Received event



Bit	Parameter Description
17	Connectionless Peripheral Broadcast Receive event
18	Connectionless Peripheral Broadcast Timeout event
19	Truncated Page Complete event
20	Peripheral Page Response Timeout event
21	Connectionless Peripheral Broadcast Channel Map Change event
22	Inquiry Response Notification event
23	Authenticated Payload Timeout Expired event
24	SAM Status Change event
25	Encryption Change event [v2]
60 to 63	Reserved for future use (used for specification development purposes)

The value with all bits set to 0 (which is the default) indicates that no events are specified.

All bits not listed in this table are reserved for future use.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Set_Event_Mask_Page_2 command succeeded.
0x01 to 0xFF	HCI_Set_Event_Mask_Page_2 command failed. See [Vol 1] Part F, Controller Error Codes for error codes and descriptions.

Event(s) generated (unless masked away):

When the HCI_Set_Event_Mask_Page_2 command has completed, an HCI_Command_Complete event shall be generated.



7.3.70 [This section is no longer used]

7.3.71 [This section is no longer used]



7.3.72 Read Flow Control Mode command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Flow_Control_Mode	0x0066		Status, Flow_Control_Mode

Description:

This command reads the value for the Flow_Control_Mode configuration parameter. See [Section 6.33](#).

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Flow_Control_Mode command succeeded.
0x01 to 0xFF	HCI_Read_Flow_Control_Mode command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Flow_Control_Mode:

Size: 1 octet

Value	Parameter Description
0x00	Packet based data flow control mode
0x01	Data block based data flow control mode
All other values	Reserved for future use

Event(s) generated (unless masked away):

When the HCI_Read_Flow_Control_Mode command has completed, an HCI_Command_Complete event shall be generated.



7.3.73 Write Flow Control Mode command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Flow_Control_Mode	0x0067	Flow_Control_Mode	Status

Description:

This command writes the value for the Flow_Control_Mode configuration parameter. See [Section 6.33](#).

Command parameters:

Flow_Control_Mode:

Size: 1 octet

Value	Parameter Description
0x00	Packet based data flow control mode (default)
0x01	Data block based data flow control mode
All other values	Reserved for future use

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Write_Flow_Control_Mode command succeeded.
0x01 to 0xFF	HCI_Write_Flow_Control_Mode command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

When the HCI_Write_Flow_Control_Mode command has completed, an HCI_Command_Complete event shall be generated. If the set fails then the Controller continues using its current mode.



7.3.74 Read Enhanced Transmit Power Level command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Enhanced_Transmit_Power_Level	0x0068	Connection_Handle, Type	Status, Connection_Handle, TX_Power_Level_GFSK, TX_Power_Level_DQPSK, TX_Power_Level_8DPSK

Description:

This command reads the values for the Enhanced_Transmit_Power_Level parameters for the specified Connection_Handle. The Connection_Handle shall be a Connection_Handle for an ACL connection.

Command parameters:

Connection_Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xXXXX	Connection_Handle Range: 0x0000 to 0x0EFF

Type: *Size: 1 octet*

Value	Parameter Description
0x00	Read Current Transmit Power Level
0x01	Read Maximum Transmit Power Level
All other values	Reserved for future use.

Return parameters:

Status: *Size: 1 octet*

Value	Parameter Description
0x00	HCI_Read_Enhanced_Transmit_Power_Level command succeeded.
0x01 to 0xFF	HCI_Read_Enhanced_Transmit_Power_Level command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

**Connection_Handle:****Size: 2 octets (12 bits meaningful)**

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

TX_Power_Level_GFSK:**Size: 1 octet**

Value	Parameter Description
0xFF	Range: -100 to 20 Units: dBm

TX_Power_Level_DQPSK:**Size: 1 octet**

Value	Parameter Description
0xFF	Range: -100 to 20 Units: dBm

TX_Power_Level_8DPSK:**Size: 1 octet**

Value	Parameter Description
0xFF	Range: -100 to 20 Units: dBm

Event(s) generated (unless masked away):

When the HCI_Read_Enhanced_Transmit_Power_Level command has completed, an HCI_Command_Complete event shall be generated.



7.3.75 [This section is no longer used]

7.3.76 [This section is no longer used]

7.3.77 [This section is no longer used]



7.3.78 Read LE Host Support command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_LE_Host_Support	0x006C		Status, LE_Supported_Host, Unused

Description:

The HCI_Read_LE_Host_Support command is used to read the LE Supported (Host) Link Manager Protocol feature bit. See [\[Vol 2\] Part C, Section 3.2](#).

The Unused parameter was previously used.

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_LE_Host_Support command succeeded.
0x01 to 0xFF	HCI_Read_LE_Host_Support command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

LE_Supported_Host:

Size: 1 octet

Value	Parameter Description
0xXX	LE_Supported_Host parameter, see Section 6.34 .

Unused:

Size: 1 octet

Value	Parameter Description
0x00	This value shall be returned by the Controller.
All other values	Reserved for future use

Event(s) generated (unless masked away):

When the HCI_Read_LE_Host_Support command has completed, an HCI_Command_Complete event shall be generated.



7.3.79 Write LE Host Support command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_LE_Host_Support	0x006D	LE_Supported_Host, Unused	Status

Description:

The HCI_Write_LE_Host_Support command is used to set the LE Supported (Host) Link Manager Protocol feature bit. See [\[Vol 2\] Part C, Section 3.2](#).

The default value for this feature bit shall be disabled. When LE_Supported_Host is set to enabled the bit in LMP features mask indicating support for LE Support (Host) shall be set.

The Unused parameter was previously used.

Command parameters:

LE_Supported_Host:

Size: 1 octet

Value	Parameter Description
0xXX	LE_Supported_Host parameter. See Section 6.34

Unused:

Size: 1 octet

Value	Parameter Description
0xXX	This value shall be ignored by the Controller.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Write_LE_Host_Support command succeeded.
0x01 to 0xFF	HCI_Write_LE_Host_Support command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

When the HCI_Write_LE_Host_Support command has completed, an HCI_Command_Complete event shall be generated.



7.3.80 Set MWS Channel Parameters command

Command	OCF	Command Parameters	Return Parameters
HCI_Set_MWS_Channel_Parameters	0x006E	MWS_Channel_Enable, MWS_RX_Center_Frequency, MWS_TX_Center_Frequency, MWS_RX_Channel_Bandwidth, MWS_TX_Channel_Bandwidth, MWS_Channel_Type	Status

Description:

The HCI_Set_MWS_Channel_Parameters command is used to inform the Controller of the MWS channel parameters.

The MWS_Channel_Enable parameter is used to enable or disable the MWS channel. If it is set to 0x00, the remaining parameters shall be ignored.

The MWS_RX_Center_Frequency and MWS_TX_Center_Frequency parameters are used to indicate the center frequency of the MWS device's uplink (TX) and downlink (RX) channels. The uplink and downlink channel centers may be the same value or different values.

The MWS_RX_Channel_Bandwidth and MWS_TX_Channel_Bandwidth parameters are used to indicate the bandwidth, in kHz, of the MWS device's uplink and downlink channels.

The MWS_Channel_Type parameter describes the type of channel. The types are defined in [Assigned Numbers](#).

Command parameters:

MWS_Channel_Enable:

Size: 1 octet

Value	Parameter Description
0x00	MWS channel is disabled.
0x01	MWS channel is enabled.
All other values	Reserved for future use.

MWS_RX_Center_Frequency:

Size: 2 octets

Value	Parameter Description
0xFFFF	MWS RX center frequency in MHz.

**MWS_TX_Center_Frequency:****Size: 2 octets**

Value	Parameter Description
0xXXXX	MWS TX center frequency in MHz

MWS_RX_Channel_Bandwidth:**Size: 2 octets**

Value	Parameter Description
0xXXXX	MWS RX channel bandwidth in kHz.

MWS_TX_Channel_Bandwidth:**Size: 2 octets**

Value	Parameter Description
0xXXXX	MWS TX channel bandwidth in kHz.

MWS_Channel_Type:**Size: 1 octet**

Value	Parameter Description
0xXX	See Assigned Numbers .

Return parameters:**Status:****Size: 1 octet**

Value	Parameter Description
0x00	HCI_Set_MWS_Channel_Parameters command succeeded.
0x01 to 0xFF	HCI_Set_MWS_Channel_Parameters command failed. See [Vol 1] Part F, Controller Error Codes , for error codes and descriptions

Event(s) generated (unless masked away):

When the HCI_Set_MWS_Channel_Parameters command has completed, an HCI_Command_Complete event shall be generated.



7.3.81 Set External Frame Configuration command

Command	OCF	Command Parameters	Return Parameters
HCI_Set_External_Frame_Configuration	0x006F	MWS_Frame_Duration, MWS_Frame_Sync_Assert_Offset, MWS_Frame_Sync_Assert_Jitter, MWS_Num_Periods, Period_Duration[i], Period_Type[i]	Status

Description:

The HCI_Set_External_Frame_Configuration command allows the Host to specify a frame configuration for an external collocated MWS system. This frame configuration persists until overwritten with a subsequent Set_External_Frame_Configuration or until the Controller is reset.

This command can be used to allow the Controller to align the piconet clock with an external frame structure.

When the external frame structure is a multiple of 1.25 ms, it can be aligned in a stable manner with the piconet clock.

The start of the external frame structure is defined as an offset from an external frame synchronization signal. This offset is defined by the MWS_Frame_Sync_Assert_Offset parameter. The offset is represented as the time (in microseconds) from the start of the next MWS frame to the FRAME_SYNC signal.

An external frame consists of downlink periods, uplink periods and guard periods. Downlink means the collocated MWS system is receiving, thus may be interfered with by Bluetooth transmissions. Uplink means the collocated MWS system is transmitting, thus may cause interference to Bluetooth receptions. A guard period may be used by the MWS system to compensate for propagation delays; in this case it should be regarded as split equally between downlink and uplink durations.

The number of specified periods is given by MWS_Num_Periods.

The duration in microseconds of each period is defined by the Period_Duration[i] parameters.

The Period_Type[i] parameter indicates if the specified period is an uplink, downlink, bi-directional or guard period.



The sum of all `Period_Duration[i]` parameters shall be equal to the `MWS_Frame_Duration` parameter.

Upon reception of an `HCI_Set_External_Frame_Configuration` command and a `FRAME_SYNC` signal from the MWS Coexistence Logical Interface, the Controller may compute the type 0 submap for local SAM slot maps. The Controller may then initiate the SAM set type 0 and SAM define map LMP sequences with the remote device.

Command parameters:

MWS_Frame_Duration: *Size: 2 octets*

Value	Parameter Description
0xFFFF	External frame duration in microseconds

MWS_Frame_Sync_Assert_Offset: *Size: 2 octets*

Value	Parameter Description
0xFFFF	External frame offset in microseconds (signed integer)

MWS_Frame_Sync_Assert_Jitter: *Size: 2 octets*

Value	Parameter Description
0xFFFF	External frame sync jitter in microseconds (unsigned integer)

MWS_Num_Periods: *Size: 1 octet*

Value	Parameter Description
0xFF	Number of specified periods in an external frame. Valid range: 1 to 32

Period_Duration[i]: *Size: MWS_Num_Periods * 2 octets*

Value	Parameter Description
0xFFFF	Duration of the period in microseconds

Period_Type[i]: *Size: MWS_Num_Periods * 1 octet*

Value	Parameter Description
0x00	Downlink
0x01	Uplink
0x02	Bi-Directional
0x03	Guard Period
All other values	Reserved for future use

**Return parameters:***Status:**Size: 1 octet*

Value	Parameter Description
0x00	HCI_Set_External_Frame_Configuration command succeeded.
0x01 to 0xFF	HCI_Set_External_Frame_Configuration command failed. See [Vol 1] Part F, Controller Error Codes , for error codes and descriptions

Event(s) generated (unless masked away):

When the HCI_Set_External_Frame_Configuration command has completed, an HCI_Command_Complete event shall be generated.



7.3.82 Set MWS Signaling command

Command	OCF	Command Parameters	Return Parameters
HCI_Set_- MWS_Signaling	0x0070	MWS_RX_Assert_Offset, MWS_RX_Assert_Jitter, MWS_RX_Deassert_Offset, MWS_RX_Deassert_Jitter, MWS_TX_Assert_Offset, MWS_TX_Assert_Jitter, MWS_TX_Deassert_Offset, MWS_TX_Deassert_Jitter, MWS_Pattern_Assert_Offset, MWS_Pattern_Assert_Jitter, MWS_Inactivity_Duration_ Assert_Offset, MWS_Inactivity_Duration_ Assert_Jitter, MWS_Scan_Frequency_ Assert_Offset, MWS_Scan_Frequency_ Assert_Jitter, MWS_Priority_Assert_ Offset_Request	Status, Bluetooth_RX_Priority_- Assert_Offset, Bluetooth_RX_Priority_- Assert_Jitter, Bluetooth_RX_Priority_- Deassert_Offset, Bluetooth_RX_Priority_- Deassert_Jitter, 802_RX_Priority_Assert_- Offset, 802_RX_Priority_Assert_- Jitter, 802_RX_Priority_Deassert_- Offset, 802_RX_Priority_Deassert_- Jitter, Bluetooth_TX_On_Assert_- Offset, Bluetooth_TX_On_Assert_- Jitter, Bluetooth_TX_On_- Deassert_Offset, Bluetooth_TX_On_- Deassert_Jitter, 802_TX_On_Assert_Offset, 802_TX_On_Assert_Jitter, 802_TX_On_Deassert_- Offset, 802_TX_On_Deassert_Jitter

Description:

The HCI_Set_MWS_Signaling command is used to inform the Bluetooth Controller of the MWS signaling interface logical layer parameters.

All signals are defined in [\[Vol 7\] Part A](#).

Command parameters:

MWS_RX_Assert_Offset:

Size: 2 octets

Value	Parameter Description
0xXXXX	MWS_RX signal assert offset in microseconds (signed integer).

*MWS_RX_Assert_Jitter:**Size: 2 octets*

Value	Parameter Description
0xXXXX	MWS_RX signal assert jitter in microseconds (unsigned integer).

*MWS_RX_Deassert_Offset:**Size: 2 octets*

Value	Parameter Description
0xXXXX	MWS_RX signal de-assert offset in microseconds (signed integer).

*MWS_RX_Deassert_Jitter:**Size: 2 octets*

Value	Parameter Description
0xXXXX	MWS_RX signal de-assert jitter in microseconds (unsigned integer).

*MWS_TX_Assert_Offset:**Size: 2 octets*

Value	Parameter Description
0xXXXX	MWS_TX signal assert offset in microseconds (signed integer).

*MWS_TX_Assert_Jitter:**Size: 2 octets*

Value	Parameter Description
0xXXXX	MWS_TX signal assert jitter in microseconds (unsigned integer).

*MWS_TX_Deassert_Offset:**Size: 2 octets*

Value	Parameter Description
0xXXXX	MWS_TX signal de-assert offset in microseconds (signed integer).

*MWS_TX_Deassert_Jitter:**Size: 2 octets*

Value	Parameter Description
0xXXXX	MWS_TX signal de-assert jitter in microseconds (unsigned integer).

*MWS_Pattern_Assert_Offset:**Size: 2 octets*

Value	Parameter Description
0xXXXX	MWS_PATTERN signal assert offset in microseconds (signed integer).

*MWS_Pattern_Assert_Jitter:**Size: 2 octets*

Value	Parameter Description
0xXXXX	MWS_PATTERN signal assert jitter in microseconds (unsigned integer).

*MWS_Inactivity_Duration_Assert_Offset:**Size: 2 octets*

Value	Parameter Description
0xFFFF	MWS_INACTIVITY_DURATION signal assert offset in microseconds (signed integer).

*MWS_Inactivity_Duration_Assert_Jitter:**Size: 2 octets*

Value	Parameter Description
0xFFFF	MWS_INACTIVITY_DURATION signal assert jitter in microseconds (unsigned integer).

*MWS_Scan_Frequency_Assert_Offset:**Size: 2 octets*

Value	Parameter Description
0xFFFF	MWS_SCAN_FREQUENCY signal assert offset in microseconds (signed integer).

*MWS_Scan_Frequency_Assert_Jitter:**Size: 2 octets*

Value	Parameter Description
0xFFFF	MWS_SCAN_FREQUENCY signal assert jitter in microseconds (unsigned integer).

*MWS_Priority_Assert_Offset_Request:**Size: 2 octets*

Value	Parameter Description
0xFFFF	Minimum advance notification from the beginning of an MWS Uplink period in microseconds (unsigned integer) before which the BLUETOOTH_RX_PRI or 802_RX_PRI signal shall be asserted to be recognized by the MWS.

Return parameters:*Status:**Size: 1 octet*

Value	Parameter Description
0x00	HCI_Set_MWS_Signaling command succeeded.
0x01 to 0xFF	HCI_Set_MWS_Signaling command failed. See [Vol 1] Part F, Controller Error Codes , for error codes and descriptions

*Bluetooth_RX_Priority_Assert_Offset:**Size: 2 octets*

Value	Parameter Description
0xFFFF	BLUETOOTH_RX_PRI signal assert offset in microseconds (signed integer).

**Bluetooth_RX_Priority_Assert_Jitter:****Size: 2 octets**

Value	Parameter Description
0xXXXX	BLUETOOTH_RX_PRI signal assert jitter in microseconds (unsigned integer).

Bluetooth_RX_Priority_Deassert_Offset:**Size: 2 octets**

Value	Parameter Description
0xXXXX	BLUETOOTH_RX_PRI signal de-assert offset in microseconds (signed integer).

Bluetooth_RX_Priority_Deassert_Jitter:**Size: 2 octets**

Value	Parameter Description
0xXXXX	BLUETOOTH_RX_PRI signal de-assert jitter in microseconds (unsigned integer).

802_RX_Priority_Assert_Offset:**Size: 2 octets**

Value	Parameter Description
0xXXXX	802_RX_PRI signal assert offset in microseconds (signed integer).

802_RX_Priority_Assert_Jitter:**Size: 2 octets**

Value	Parameter Description
0xXXXX	802_RX_PRI signal assert jitter in microseconds (unsigned integer).

802_RX_Priority_Deassert_Offset:**Size: 2 octets**

Value	Parameter Description
0xXXXX	802_RX_PRI signal de-assert offset in microseconds (signed integer).

802_RX_Priority_Deassert_Jitter:**Size: 2 octets**

Value	Parameter Description
0xXXXX	802_RX_PRI signal de-assert jitter in microseconds (unsigned integer).

Bluetooth_TX_On_Assert_Offset:**Size: 2 octets**

Value	Parameter Description
0xXXXX	BLUETOOTH_TX_ON signal assert offset in microseconds (signed integer).

*Bluetooth_TX_On_Assert_Jitter:**Size: 2 octets*

Value	Parameter Description
0xFFFF	BLUETOOTH_TX_ON signal assert jitter in microseconds (unsigned integer).

*Bluetooth_TX_On_Deassert_Offset:**Size: 2 octets*

Value	Parameter Description
0xFFFF	BLUETOOTH_TX_ON signal de-assert offset in microseconds (signed integer).

*Bluetooth_TX_On_Deassert_Jitter:**Size: 2 octets*

Value	Parameter Description
0xFFFF	BLUETOOTH_TX_ON signal de-assert jitter in microseconds (unsigned integer).

*802_TX_On_Assert_Offset:**Size: 2 octets*

Value	Parameter Description
0xFFFF	802_TX_ON signal assert offset in microseconds (signed integer).

*802_TX_On_Assert_Jitter:**Size: 2 octets*

Value	Parameter Description
0xFFFF	802_TX_ON signal assert jitter in microseconds (unsigned integer).

*802_TX_On_Deassert_Offset:**Size: 2 octets*

Value	Parameter Description
0xFFFF	802_TX_ON signal de-assert offset in microseconds (signed integer).

*802_TX_On_Deassert_Jitter:**Size: 2 octets*

Value	Parameter Description
0xFFFF	802_TX_ON signal de-assert jitter in microseconds (unsigned integer).

Event(s) generated (unless masked away):

When the HCI_Set_MWS_Signaling command has completed, an HCI_Command_Complete event shall be generated.



7.3.83 Set MWS Transport Layer command

Command	OCF	Command Parameters	Return Parameters
HCI_Set_MWS_Transport_Layer	0x0071	Transport_Layer, To_MWS_Baud_Rate, From_MWS_Baud_Rate	Status

Description:

The HCI_Set_MWS_Transport_Layer command configures the transport layer between the Bluetooth Controller and MWS device.

Command parameters:

Transport_Layer: *Size: 1 octet*

Value	Parameter Description
0xXX	See Assigned Numbers .

To_MWS_Baud_Rate: *Size: 4 octets*

Value	Parameter Description
0XXXXXXXXX	Baud rate in the Bluetooth to MWS direction in Baud.

From_MWS_Baud_Rate: *Size: 4 octets*

Value	Parameter Description
0XXXXXXXXX	Baud rate in the MWS to Bluetooth direction in Baud.

Return parameters:

Status: *Size: 1 octet*

Value	Parameter Description
0x00	HCI_Set_MWS_Transport_Layer command succeeded.
0x01 to 0xFF	HCI_Set_MWS_Transport_Layer command failed. See [Vol 1] Part F, Controller Error Codes , for error codes and descriptions

Event(s) generated (unless masked away):

When the HCI_Set_MWS_Transport_Layer command has completed, an HCI_Command_Complete event shall be generated.



7.3.84 Set MWS Scan Frequency Table command

Command	OCF	Command Parameters	Return Parameters
HCI_Set_MWS_Scan_Frequency_Table	0x0072	Num_Scan_Frequencies, Scan_Frequency_Low[i], Scan_Frequency_High[i]	Status

Description:

The HCI_Set_MWS_Scan_Frequency_Table command configures the MWS scan frequency table in the Controller.

The Num_Scan_Frequencies parameter indicates the number of MWS scan frequencies to be set. A Controller shall support at least 8 table entries.

The Scan_Frequency_Low[i] and Scan_Frequency_High[i] parameters indicate the lower and upper edges for each scan frequency.

Command parameters:

Num_Scan_Frequencies:

Size: 1 octet

Value	Parameter Description
N	Number of MWS scan frequencies to be set in the table.

Scan_Frequency_Low[i]:

Size: Num_Scan_Frequencies × 2 octets

Value	Parameter Description
0xXXXX	Lower edge of the MWS scan frequency in MHz.

Scan_Frequency_High[i]:

Size: Num_Scan_Frequencies × 2 octets

Value	Parameter Description
0xXXXX	Upper edge of the MWS scan frequency in MHz.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Set_MWS_Scan_Frequency_Table command succeeded.
0x01 to 0xFF	HCI_Set_MWS_Scan_Frequency_Table command failed. See [Vol 1] Part F, Controller Error Codes , for error codes and descriptions

**Event(s) generated (unless masked away):**

When the HCI_Set_MWS_Scan_Frequency_Table command has completed, an HCI_Command_Complete event shall be generated.



7.3.85 Set MWS_PATTERN Configuration command

Command	OCF	Command Parameters	Return Parameters
HCI_Set_MWS_PATTERN_Configuration	0x0073	MWS_Pattern_Index, MWS_Pattern_Num_Intervals, MWS_Pattern_Interval_Duration[i], MWS_Pattern_Interval_Type[i]	Status

Description:

The HCI_Set_MWS_PATTERN_Configuration command is used by the Host to specify, in conjunction with the HCI_Set_External_Frame_Configuration command, local MWS_PATTERN parameters for an external collocated system.

An MWS_PATTERN configuration shall persist until overwritten by a subsequent Set_MWS_PATTERN_Configuration. All MWS_PATTERN configurations are deleted when an HCI_Set_External_Frame_Configuration command is received or when the Controller is reset.

The sum of the MWS_Pattern_Interval_Duration parameters shall be an integer multiple of the length of a frame as defined by the most recent HCI_Set_External_Frame_Configuration command.

If any interval with type 4 either has an MWS_Pattern_Interval_Duration greater than the length of a frame or the sum of the MWS_Pattern_Interval_Duration parameters for the previous intervals is not a multiple of the length of the frame, the Controller shall return the error code *Invalid HCI Command Parameters* (0x12).

Upon reception of an HCI_Set_MWS_PATTERN_Configuration command, the Controller may compute the local SAM slot map with SAM_Index equal to MWS_Pattern_Index. If the SAM slot map does not exist, it should be created; if the SAM slot map already exists, its parameters should be replaced. The Controller may then initiate the SAM define map LMP sequence with the remote device.

Upon reception of an MWS_PATTERN signal, with a value other than 3, from the MWS Coexistence Logical Interface (see [\[Vol 7\] Part A](#)), the Controller should check the MWS_PATTERN value against the SAM_Index of those SAM slot maps that have been configured by previous HCI_Set_MWS_PATTERN_Configuration commands. It should then take the following course of action:

1. If MWS_PATTERN does not match any configured SAM slot map, it should take no further action.
2. If MWS_PATTERN matches an available SAM slot map that is already active or is being activated, it should take no further action (i.e. let the current or pending active SAM slot map continue).



3. If MWS_PATTERN matches an available SAM slot map that is neither active nor is being activated, then:

- a) If the SAM slot map has been activated previously using the LMP_SAM_SET_TYPE0 (if relevant) and LMP_SAM_DEFINE_MAP LMP sequences, the Controller should start the SAM switch LMP sequence to activate the matched SAM slot map;
- b) Otherwise the Controller should start or complete the SAM set type 0 (if relevant), SAM define map, and SAM switch LMP sequences to activate the matched SAM slot map.

Command parameters:

MWS_Pattern_Index:

Size: 1 octet

Value	Parameter Description
0xXX	Index of the MWS_PATTERN instance to be configured. Range: 0 to 2.

MWS_Pattern_Num_Intervals:

Size: 1 octet

Value	Parameter Description
0xXX	The number of intervals in the following arrays.

MWS_Pattern_Interval_Duration[i]:

Size: MWS_Pattern_Num_Intervals × 2 octets

Value	Parameter Description
0XXXXX	The duration of this Bluetooth activity interval in microseconds.

MWS_Pattern_Interval_Type[i]: *Size: MWS_Pattern_Num_Intervals × 1 octet*

Value	Parameter Description
0x00	Neither transmission nor reception is allowed in this interval.
0x01	Transmission is allowed in this interval.
0x02	Reception is allowed in this interval.
0x03	Both transmission and reception are allowed in this interval.
0x04	Interval for the MWS frame as defined by the HCI_Set_External_Frame_Configuration command.
All other values	Reserved for future use



Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Set_MWS_PATTERN_Configuration command succeeded
0x01 to 0xFF	HCI_Set_MWS_PATTERN_Configuration command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

When the HCI_Set_MWS_PATTERN_Configuration command has completed, an HCI_Command_Complete event shall be generated.



7.3.86 Set Reserved LT_ADDR command

Command	OCF	Command Parameters	Return Parameters
HCI_Set_Reserved_LT_ADDR	0x0074	LT_ADDR	Status, LT_ADDR

Description:

The HCI_Set_Reserved_LT_ADDR command allows the Host to request that the BR/EDR Controller reserve a specific LT_ADDR for Connectionless Peripheral Broadcast.

If the LT_ADDR indicated in the LT_ADDR parameter is already in use by the BR/EDR Controller, it shall return the *Connection Already Exists* (0x0B) error code. If the LT_ADDR indicated in the LT_ADDR parameter is out of range, the Controller shall return the *Invalid HCI Command Parameters* (0x12) error code. If the command succeeds, then the reserved LT_ADDR shall be used when issuing subsequent HCI_Set_Connectionless_Peripheral_Broadcast_Data and HCI_Set_Connectionless_Peripheral_Broadcast commands.

To ensure that the reserved LT_ADDR is not already allocated, it is recommended that this command be issued at some point after HCI_Reset is issued but before page scanning is enabled or paging is initiated.

Command parameters:

LT_ADDR:

Size: 1 octet

Value	Parameter Description
0x01 to 0x07	LT_ADDR to reserve for Connectionless Peripheral Broadcast
All other values	Reserved for future use

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Set_Reserved_LT_ADDR command succeeded.
0x01 to 0xFF	HCI_Set_Reserved_LT_ADDR command failed. See [Vol 1] Part F, Controller Error Codes , for error codes and descriptions.

**LT_ADDR:****Size: 1 octet**

Value	Parameter Description
0x01 to 0x07	LT_ADDR reserved for Connectionless Peripheral Broadcast. This parameter shall have the same value as the Command Parameter LT_ADDR.
All other values	Reserved for future use

Event(s) generated (unless masked away):

When the HCI_Set_Reserved_LT_ADDR command has completed, an HCI_Command_Complete event shall be generated.



7.3.87 Delete Reserved LT_ADDR command

Command	OCF	Command Parameters	Return Parameters
HCI_Delete_Reserved_LT_ADDR	0x0075	LT_ADDR	Status, LT_ADDR

Description:

The HCI_Delete_Reserved_LT_ADDR command requests that the BR/EDR Controller cancel the reservation for a specific LT_ADDR reserved for the purposes of Connectionless Peripheral Broadcast.

If the LT_ADDR indicated in the LT_ADDR parameter is not reserved by the BR/EDR Controller, it shall return the *Unknown Connection Identifier* (0x02) error code.

If Connectionless Peripheral Broadcast mode is still active, then the Controller shall return the *Command Disallowed* (0x0C) error code.

Command parameters:

LT_ADDR:

Size: 1 octet

Value	Parameter Description
0x01 to 0x07	LT_ADDR currently reserved for Connectionless Peripheral Broadcast and for which reservation is to be cancelled
All other values	Reserved for future use

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Delete_Reserved_LT_ADDR command succeeded.
0x01 to 0xFF	HCI_Delete_Reserved_LT_ADDR command failed. See [Vol 1] Part F, Controller Error Codes , for error codes and descriptions.

LT_ADDR:

Size: 1 octet

Value	Parameter Description
0x01 to 0x07	LT_ADDR whose reservation the Host has requested to cancel.
All other values	Reserved for future use

Event(s) generated (unless masked away):

When the HCI_Delete_Reserved_LT_ADDR command has completed, an HCI_Command_Complete event shall be generated.



7.3.88 Set Connectionless Peripheral Broadcast Data command

Command	OCF	Command Parameters	Return Parameters
HCI_Set_Connectionless_Peripheral_Broadcast_Data	0x0076	LT_ADDR, Fragment, Data_Length, Data	Status, LT_ADDR

Description:

The HCI_Set_Connectionless_Peripheral_Broadcast_Data command provides the ability for the Host to set Connectionless Peripheral Broadcast data in the BR/EDR Controller. This command may be issued at any time after an LT_ADDR has been reserved regardless of whether Connectionless Peripheral Broadcast mode has been enabled or disabled by the Enable parameter in the HCI_Set_Connectionless_Peripheral_Broadcast command. If the command is issued without the LT_ADDR reserved, the *Unknown Connection Identifier* (0x02) error code shall be returned.

If Connectionless Peripheral Broadcast mode is disabled, this data shall be kept by the BR/EDR Controller and used once Connectionless Peripheral Broadcast mode is enabled. If Connectionless Peripheral Broadcast mode is enabled, and this command is successful, this data will be sent starting with the next Connectionless Peripheral Broadcast instant.

The Data_Length field may be zero, in which case no data needs to be provided.

The Host may fragment the data using the Fragment field in the command. If the combined length of the fragments exceeds the capacity of the largest allowed packet size specified in the HCI_Set_Connectionless_Peripheral_Broadcast command, all fragments associated with the data being assembled shall be discarded and the *Invalid HCI Command Parameters* error code (0x12) shall be returned.

Command parameters:

LT_ADDR:

Size: 1 octet

Value	Parameter Description
0x01 to 0x07	LT_ADDR on which to send Connectionless Peripheral Broadcast data
All other values	Reserved for future use

**Fragment:****Size: 1 octet**

Value	Parameter Description
0x00	Continuation fragment
0x01	Starting fragment
0x02	Ending fragment
0x03	No fragmentation (single fragment)
All other values	Reserved for future use

Data_Length:**Size: 1 octet**

Value	Parameter Description
0xXX	Length of the Data field

Data:**Size: Data_Length octets**

Value	Parameter Description
Variable	Data to send in future Connectionless Peripheral Broadcast packets. This data will be repeated in future Connectionless Peripheral Broadcast instants until new data is provided

Return parameters:**Status:****Size: 1 octet**

Value	Parameter Description
0x00	HCI_Set_Connectionless_Peripheral_Broadcast_Data command succeeded.
0x01 to 0xFF	HCI_Set_Connectionless_Peripheral_Broadcast_Data command failed. See [Vol 1] Part F, Controller Error Codes , for error codes and descriptions

LT_ADDR:**Size: 1 octet**

Value	Parameter Description
0x01 to 0x07	LT_ADDR on which Connectionless Peripheral Broadcast data will be sent
All other values	Reserved for future use

Event(s) generated (unless masked away):

When the HCI_Set_Connectionless_Peripheral_Broadcast_Data command has completed, an HCI_Command_Complete event shall be generated.



7.3.89 Read Synchronization Train Parameters command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Synchronization_Train_Parameters	0x0077		Status, Sync_Train_Interval, Sync_Train_Timeout, Service_Data

Description:

The HCI_Read_Synchronization_Train_Parameters command returns the currently configured values for the Synchronization Train functionality in the Central's BR/EDR Controller. This command may be issued at any time.

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Synchronization_Train_Parameters command succeeded.
0x01 to 0xFF	HCI_Read_Synchronization_Train_Parameters command failed. See [Vol 1] Part F, Controller Error Codes , for error codes and descriptions.

Sync_Train_Interval:

Size: 2 octets

Value	Parameter Description
0xFFFF	Interval in slots between consecutive Synchronization Train events on the same channel. Range: 0x0020 to 0xFFFFE; only even values are valid

Sync_Train_Timeout:

Size: 4 octets

Value	Parameter Description
0xFFFFFFFF	Duration in slots to continue sending the synchronization train Range: 0x00000002 to 0xFFFFFFFF; only even values are valid

**Service_Data:****Size: 1 octet**

Value	Parameter Description
0xXX	Host provided value included in Synchronization Train packet, octet 27; see [Vol 2] Part B, Table 8.11 .

Event(s) generated (unless masked away):

When the BR/EDR Controller receives the HCI_Read_Synchronization_Train_Parameters command, it shall send an HCI_Command_Complete event to the Host.



7.3.90 Write Synchronization Train Parameters command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Synchronization_Train_Parameters	0x0078	Interval_Min, Interval_Max, Sync_Train_Timeout, Service_Data	Status, Sync_Train_Interval

Description:

The HCI_Write_Synchronization_Train_Parameters command configures the Synchronization Train functionality in the BR/EDR Controller. This command may be issued at any time.

Note: The AFH_Channel_Map used in the Synchronization Train packets is configured by the HCI_Set_AFH_Host_Channel_Classification command and the local channel classification in the BR/EDR Controller.

Interval_Min and Interval_Max specify the allowed range of Sync_Train_Interval. Refer to [\[Vol 2\] Part B, Section 2.7.2](#) for a detailed description of Sync_Train_Interval. The BR/EDR Controller shall select an interval from this range and return it in Sync_Train_Interval. If the Controller is unable to select a value from this range, it shall return the *Invalid HCI Command Parameters* (0x12) error code.

Once started (via the HCI_Start_Synchronization_Train command) the Synchronization Train will continue until Sync_Train_Timeout slots have passed or Connectionless Peripheral Broadcast has been disabled.

Command parameters:

Interval_Min:

Size: 2 octets

Value	Parameter Description
0xXXXX	Minimum value allowed for the interval Sync_Train_Interval in slots. Range: 0x0020 to 0xFFFFE; only even values are valid

Interval_Max:

Size: 2 octets

Value	Parameter Description
0xXXXX	Maximum value allowed for the interval Sync_Train_Interval in slots. Range: 0x0020 to 0xFFFFE; only even values are valid

**Sync_Train_Timeout:****Size: 4 octets**

Value	Parameter Description
0xFFFFFFFF	Duration in slots to continue sending the synchronization train Range: 0x00000002 to 0x07FFFFFFE; only even values are valid

Service_Data:**Size: 1 octet**

Value	Parameter Description
0xFF	Host provided value to be included in octet 27 of the Synchronization Train packet payload body; see [Vol 2] Part B, Table 8.11.

Return parameters:**Status:****Size: 1 octet**

Value	Parameter Description
0x00	HCI_Write_Synchronization_Train_Parameters command succeeded.
0x01 to 0xFF	HCI_Write_Synchronization_Train_Parameters command failed. See [Vol 1] Part F, Controller Error Codes, for error codes and descriptions.

Sync_Train_Interval:**Size: 2 octets**

Value	Parameter Description
0xFFFF	Interval in slots between consecutive Synchronization Train packets on the same channel. Range: 0x0020 to 0xFFFFE; only even values are valid

Event(s) generated (unless masked away):

When the BR/EDR Controller receives the HCI_Write_Synchronization_Train_Parameters command, it shall send an HCI_Command_Complete event to the Host.



7.3.91 Read Secure Connections Host Support command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Secure_Connections_Host_Support	0x0079		Status, Secure_Connections_Host_Support

Description:

This command reads the Secure_Connections_Host_Support parameter in the BR/EDR Controller. When Secure Connections Host Support is set to 'enabled' the Controller uses the enhanced reporting mechanisms for the Encryption_Enabled parameter in the HCI_Encryption_Change event (see [Section 7.7.8](#)) and the Key_Type parameter in the HCI_Link_Key_Notification event (see [Section 7.7.24](#)).

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Secure_Connections_Host_Support command succeeded.
0x01 to 0xFF	HCI_Read_Secure_Connections_Host_Support command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Secure_Connections_Host_Support:

Size: 1 octet

Value	Parameter Description
0x00	Secure_Connections_Host_Support is 'disabled'. Host does not support Secure Connections (default)
0x01	Secure_Connections_Host_Support is 'enabled'. Host supports Secure Connections
All other values	Reserved for future use

Event(s) generated (unless masked away):

When the HCI_Read_Secure_Connections_Host_Support command has completed, an HCI_Command_Complete event shall be generated.



7.3.92 Write Secure Connections Host Support command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Secure_Connections_Host_Support	0x007A	Secure_Connections_Host_Support	Status

Description:

This command writes the Secure_Connections_Host_Support parameter in the BR/EDR Controller. When Secure Connections Host Support is set to 'enabled' the Controller shall use the enhanced reporting mechanisms for the Encryption_Enabled parameter in the HCI_Encryption_Change event (see [Section 7.7.8](#)) and the Key_Type parameter in the HCI_Link_Key_Notification event (see [Section 7.7.24](#)). If the Host issues this command while the Controller is paging, has page scanning enabled, or has an ACL connection, the Controller shall return the error code *Command Disallowed* (0x0C).

The Link Manager Secure Connections (Host Support) feature bit shall be set to the Secure_Connections_Host_Support parameter. The default value for Secure_Connections_Host_Support shall be 'disabled.' When Secure_Connections_Host_Support is set to 'enabled,' the bit in the LMP features mask indicating support for Secure Connections (Host Support) shall be set to enabled in subsequent responses to an LMP_FEATURES_REQ from a remote device.

Command parameters:

Secure_Connections_Host_Support:

Size: 1 octet

Value	Parameter Description
0x00	Secure_Connections_Host_Support is 'disabled'. Host does not support Secure Connections (default)
0x01	Secure_Connections_Host_Support is 'enabled'. Host supports Secure Connections
All other values	Reserved for future use

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Write_Secure_Connections_Host_Support command succeeded.
0x01 to 0xFF	HCI_Write_Secure_Connections_Host_Support command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

**Event(s) generated (unless masked away):**

When the HCI_Write_Secure_Connections_Host_Support command has completed, an HCI_Command_Complete event shall be generated.



7.3.93 Read Authenticated Payload Timeout command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Authenticated_Payload_Timeout	0x007B	Connection_Handle	Status, Connection_Handle, Authenticated_Payload_Timeout

Description:

This command reads the Authenticated_Payload_Timeout (*authenticatedPayloadTO*, see [Vol 2] Part B, Appendix B for BR/EDR connections and [Vol 6] Part B, Section 5.4 for LE connections) parameter in the Controller on the specified Connection_Handle.

When the Connection_Handle identifies a BR/EDR synchronous connection, the Controller shall return the error code *Command Disallowed* (0x0C).

Command parameters:

Connection_Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Authenticated_Payload_Timeout command succeeded.
0x01 to 0xFF	HCI_Read_Authenticated_Payload_Timeout command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Connection_Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

**Authenticated_Payload_Timeout:****Size: 2 octets**

Value	Parameter Description
N = 0xXXXX	Maximum amount of time specified between packets authenticated by a MIC. Default = 0x0BB8 (30 s) Range: 0x0001 to 0xFFFF Time = N * 10 ms Time Range: 10 ms to 655,350 ms

Event(s) generated (unless masked away):

When the HCI_Read_Authenticated_Payload_Timeout command has completed, an HCI_Command_Complete event shall be generated.



7.3.94 Write Authenticated Payload Timeout command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Authenticated_Payload_Timeout	0x007C	Connection_Handle, Authenticated_Payload_Timeout	Status, Connection_Handle

Description:

This command writes the Authenticated_Payload_Timeout (*authenticatedPayloadTO*, see [Vol 2] Part B, Appendix B and [Vol 6] Part B, Section 5.4 for the LE connection) parameter in the Controller for the specified Connection_Handle.

When the Connection_Handle identifies a BR/EDR ACL connection:

- If the connection is in Sniff mode, the Authenticated_Payload_Timeout shall be equal to or greater than T_{sniff} .
- If the connection is in Sniff Subrating mode, the Authenticated_Payload_Timeout shall be equal to or greater than $(\text{max subrate}) \times T_{\text{sniff}}$.
- If the connection is in Hold mode, the Authenticated_Payload_Timeout shall be equal to or greater than the *holdTO* value.

When the Connection_Handle identifies a BR/EDR synchronous connection, this command shall be rejected with the error code *Command Disallowed* (0x0C).

When the Connection_Handle identifies an LE connection, the Authenticated_Payload_Timeout shall be equal to or greater than $\text{connInterval} \times \text{connSubrateFactor} \times (1 + \text{connPeripheralLatency})$.

When the Connection_Handle is associated with an ACL connection, the Link Manager will use this parameter to determine when to use the LMP ping sequence.

When the Connection_Handle is associated with an LE connection, the Link Layer will use this parameter to determine when to use the LE ping sequence.

**Command parameters:***Connection_Handle:**Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xXXXX	Connection_Handle Range: 0x0000 to 0x0EFF

*Authenticated_Payload_Timeout:**Size: 2 octets*

Value	Parameter Description
N = 0xXXXX	Maximum amount of time specified between packets authenticated by a valid MIC. Range: 0x0001 to 0xFFFF Time = N * 10 ms Time Range: 10 ms to 655,350 ms

Return parameters:*Status:**Size: 1 octet*

Value	Parameter Description
0x00	HCI_Write_Authenticated_Payload_Timeout command succeeded.
0x01 to 0xFF	HCI_Write_Authenticated_Payload_Timeout command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

*Connection_Handle:**Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xXXXX	Connection_Handle Range: 0x0000 to 0x0EFF

Event(s) generated (unless masked away):

When the HCI_Write_Authenticated_Payload_Timeout command has completed, an HCI_Command_Complete event shall be generated.



7.3.95 Read Local OOB Extended Data command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Local_OOB_Extended_Data	0x007D		Status, C_192, R_192, C_256, R_256

Description:

This command obtains the Secure Simple Pairing Hash C_192, Randomizer R_192, Hash C_256, and Randomizer R_256, which are intended to be transferred to a remote device using an OOB mechanism. The BR/EDR Controller shall create new values for C_192, R_192, C_256, and R_256 for each invocation of this command. Each random number (R_192 and R_256) shall be created according to [\[Vol 2\] Part H, Section 2](#).

If the Host issues this command before enabling Secure Connections Host Support, the Controller shall return the error code *Command Disallowed* (0x0C).

Note: Each OOB transfer will have unique C_192, R_192, C_256, and R_256 values.

After each OOB transfer this command shall be used to obtain a new set of values for the next OOB transfer.

Note: The Controller keeps information used to generate these values for later use in the Secure Simple Pairing process. If the BR/EDR Controller is powered off or reset then this information is lost and the values obtained before the power off or reset are invalid.

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Local_OOB_Extended_Data command succeeded.
0x01 to 0xFF	HCI_Read_Local_OOB_Extended_Data command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

**C_192:****Size: 16 octets**

Value	Parameter Description
0XXXXXXXXXXXXX XXXXXXXXXXXXX XXXXXXXXXXXXX	Secure Simple Pairing Hash C derived from the P-192 public key.

R_192:**Size: 16 octets**

Value	Parameter Description
0XXXXXXXXXXXXX XXXXXXXXXXXXX XXXXXXXXXXXXX	Secure Simple Pairing Randomizer associated with the P-192 public key.

C_256:**Size: 16 octets**

Value	Parameter Description
0XXXXXXXXXXXXX XXXXXXXXXXXXX XXXXXXXXXXXXX	Secure Simple Pairing Hash C derived from the P-256 public key.

R_256:**Size: 16 octets**

Value	Parameter Description
0XXXXXXXXXXXXX XXXXXXXXXXXXX XXXXXXXXXXXXX	Secure Simple Pairing Randomizer associated with the P-256 public key.

Event(s) generated (unless masked away):

When the HCI_Read_Local_OOB_Extended_Data command has completed, an HCI_Command_Complete event shall be generated.



7.3.96 Read Extended Page Timeout command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Extended_Page_Timeout	0x007E		Status, Extended_Page_Timeout

Description:

The HCI_Read_Extended_Page_Timeout command will read the value for the Extended_Page_Timeout configuration parameter. See [Section 6.41](#).

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Extended_Page_Timeout command succeeded.
0x01 to 0x0F	HCI_Read_Extended_Page_Timeout command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Extended_Page_Timeout:

Size: 2 octets

Value	Parameter Description
0xXXXX	Extended Page Timeout measured in number of Baseband slots. Interval Length = $N * 0.625 \text{ ms}$ (1 Baseband slot) Range: 0x0000 (default) to 0xFFFF Time Range: 0 to 40.9 s

Event(s) generated (unless masked away):

When the HCI_Read_Extended_Page_Timeout command has completed, an HCI_Command_Complete event shall be generated.



7.3.97 Write Extended Page Timeout command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Extended_Page_Timeout	0x007F	Extended_Page_Timeout	Status

Description:

The HCI_Write_Extended_Page_Timeout command will write the value for the Extended_Page_Timeout configuration parameter. See [Section 6.41](#).

Command parameters:

Extended_Page_Timeout:

Size: 2 octets

Value	Parameter Description
N = 0xXXXX	Extended Page Timeout measured in number of Baseband slots. Interval Length = $N * 0.625$ ms (1 Baseband slot) Range: 0x0000 (default) to 0xFFFF Time Range: 0 to 40.9 s

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Write_Extended_Page_Timeout command succeeded.
0x01 to 0x0F	HCI_Write_Extended_Page_Timeout command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

When the HCI_Write_Extended_Page_Timeout command has completed, an HCI_Command_Complete event shall be generated.



7.3.98 Read Extended Inquiry Length command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Extended_Inquiry_Length	0x0080		Status, Extended_Inquiry_Length

Description:

The HCI_Read_Extended_Inquiry_Length command will read the value for the Extended_Inquiry_Length configuration parameter. See [Section 6.42](#).

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Extended_Inquiry_Length command succeeded.
0x01 to 0x0F	HCI_Read_Extended_Inquiry_Length command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Extended_Inquiry_Length:

Size: 2 octets

Value	Parameter Description
0xXXXX	Extended_Inquiry_Length measured in number of Baseband slots. Interval Length = $N * 0.625 \text{ ms}$ (1 Baseband slot) Range: 0x0000 (default) to 0xFFFF Time Range: 0 to 40.9 s

Event(s) generated (unless masked away):

When the HCI_Read_Extended_Inquiry_Length command has completed, an HCI_Command_Complete event shall be generated.



7.3.99 Write Extended Inquiry Length command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Extended_Inquiry_Length	0x0081	Extended_Inquiry_Length	Status

Description:

The HCI_Write_Extended_Inquiry_Length command will write the value for the Extended_Inquiry_Length configuration parameter. The Extended_Inquiry_Length configuration parameter defines the maximum time after the Inquiry_Length expires that the local Link Manager may wait for a Baseband inquiry response from the remote device at a locally initiated connection attempt. If this time expires and the remote device has not responded to the inquiry at Baseband level, the inquiry will be considered to have failed.

Command parameters:

Extended_Inquiry_Length:

Size: 2 octets

Value	Parameter Description
0xXXXX	Extended_Inquiry_Length measured in number of Baseband slots. Interval Length = $N * 0.625$ ms (1 Baseband slot) Range: 0x0000 (default) to 0xFFFF Time Range: 0 to 40.9 s

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Write_Extended_Inquiry_Length command succeeded.
0x01 to 0x0F	HCI_Write_Extended_Inquiry_Length command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

When the HCI_Write_Extended_Inquiry_Length command has completed, an HCI_Command_Complete event shall be generated.



7.3.100 Set Ecosystem Base Interval command

Command	OCF	Command Parameters	Return Parameters
HCI_Set_Ecosystem_Base_Interval	0x0082	Interval	Status

Description:

The HCI_Set_Ecosystem_Base_Interval command provides a hint to the Controller specifying the base communication interval the Controller can expect current and future communications to use. The Controller can assume that future activities will use an interval that is a multiple of the hint and may use that assumption when scheduling future activities. For example, if the Host expects to set up an LE ACL with a connection interval of 15 ms and a periodic advertisement with an interval of 40 ms, it would issue this command with Interval set to 5 ms. If the Host gives a range of possible intervals for an activity, the Controller could use a value in that range that is a multiple of the hint; e.g., if it gives the range 37.5 to 42.5 ms, the Controller could use 40 ms in preference to any other value. Communications being scheduled can include, but are not limited to, (e)SCO connections, BR/EDR ACL connections in Sniff mode, LE ACL connections, periodic advertisements, CIses, and BISes.

The Host may use an interval of zero to indicate that the most recently provided hint is no longer valid.

Command parameters:

Interval:

Size: 2 octets

Value	Parameter Description
0x0000	Ignore any previous hint
N = 0xXXXX	Base interval of the ecosystem Range: 0x0002 to 0x7DF0 Time = N * 1.25 ms Time Range: 2.5 ms to 40.9 s.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Set_Ecosystem_Base_Interval command succeeded.
All other values	HCI_Set_Ecosystem_Base_Interval command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

**Event(s) generated (unless masked away):**

When the HCI_Set_Ecosystem_Base_Interval command has completed, an HCI_Command_Complete event shall be generated.



7.3.101 Configure Data Path command

Command	OCF	Command Parameters	Return Parameters
HCI_Configure_Data_Path	0x0083	Data_Path_Direction, Data_Path_ID, Vendor_Specific_Config_Length, Vendor_Specific_Config	Status

Description:

The HCI_Configure_Data_Path command is used to request the Controller to configure the data transport path in a given direction between the Controller and the Host.

The Data_Path_Direction parameter specifies the direction to be configured.

The Data_Path_ID parameter shall indicate the logical transport channel number for the non-HCI transport data path (e.g PCM interface) to be configured. The meaning of these logical transport channel numbers is vendor-specific.

The Vendor_Specific_Config parameter specifies additional vendor-specific configuration information that a Host may provide to the Controller.

If the Host issues this command with a value of Data_Path_ID that is not supported, the Controller shall return the error code *Invalid HCI Command Parameters* (0x12).

Command parameters:

Data_Path_Direction:

Size: 1 octet

Value	Parameter Description
0x00	Input (Host to Controller)
0x01	Output (Controller to Host)
All other values	Reserved for future use

Data_Path_ID:

Size: 1 octet

Value	Parameter Description
0x00	Reserved for future use
0x01 to 0xFE	Logical channel number; the meaning is vendor-specific.
0xFF	Reserved for future use

**Vendor_Specific_Config_Length:****Size: 1 octet**

Value	Parameter Description
0xXX	Length of the vendor-specific configuration data

Vendor_Specific_Config:**Size: Vendor_Specific_Config_Length octets**

Value	Parameter Description
Variable	Vendor-specific configuration data for the data path being configured

Return parameters:**Status:****Size: 1 octet**

Value	Parameter Description
0x00	HCI_Configure_Data_Path command succeeded
0x01 to 0xFF	HCI_Configure_Data_Path command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

When the HCI_Configure_Data_Path command has completed, an HCI_Command_Complete event shall be generated.



7.3.102 Set Min Encryption Key Size command

Command	OCF	Command Parameters	Return Parameters
HCI_Set_Min_Encryption_Key_Size	0x0084	Min_Encryption_Key_Size	Status

Description:

The HCI_Set_Min_Encryption_Key_Size command is used by the Host to configure the minimum encryption key size. The Controller shall not negotiate a key size smaller than this value for any subsequent connection over the BR/EDR transport. This command shall not affect any existing connections.

The Min_Encryption_Key_Size parameter specifies the new minimum encryption key size in octets. See [Vol 3] Part C, Section 5.2.2.8 for recommendations concerning key sizes.

If the Host specifies a minimum encryption key size that the Controller does not support, the Controller shall return the error code *Unsupported Feature or Parameter Value* (0x11).

Command parameters:

Min_Encryption_Key_Size:

Size: 1 octet

Value	Parameter Description
0xXX	Minimum encryption key size in octets. Range: 0x01 to 0x10 Default: vendor-specific

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Set_Min_Encryption_Key_Size command succeeded
0x01-0xFF	HCI_Set_Min_Encryption_Key_Size command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

When the HCI_Set_Min_Encryption_Key_Size command has completed, an HCI_Command_Complete event shall be generated.



7.4 INFORMATIONAL PARAMETERS

The informational parameters are fixed by the manufacturer of the Bluetooth hardware. These parameters provide information about the BR/EDR Controller and the capabilities of the Link Manager and Baseband in the BR/EDR Controller. The Host device cannot modify any of these parameters.

For Informational Parameters commands, the OGF is defined as 0x04.

7.4.1 Read Local Version Information command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Local_Version_Information	0x0001		Status, HCI_Version, HCI_Subversion, LMP_Version, Company_Identifier, LMP_Subversion

Description:

This command reads the values for the version information for the local Controller.

The HCI_Version information defines the version information of the HCI layer. The LMP_Version information defines the version of the LMP. The Company_Identifier information indicates the manufacturer of the local device.

The HCI_Subversion and LMP_Subversion are vendor-specific.

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Local_Version_Information command succeeded.
0x01 to 0xFF	HCI_Read_Local_Version_Information command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

**HCI_Version:****Size: 1 octet**

Value	Parameter Description
0xXX	Version of the HCI Specification supported by the Controller See Assigned Numbers

HCI_Subversion:**Size: 2 octets**

Value	Parameter Description
0XXXXX	Revision of the HCI implementation in the Controller. This value is vendor-specific.

LMP_Version:**Size: 1 octet**

Value	Parameter Description
0xXX	Version of the Current LMP supported by the Controller. See Assigned Numbers

Company_Identifier:**Size: 2 octets**

Value	Parameter Description
0XXXXX	Company identifier for the manufacturer of the Controller. See Assigned Numbers

LMP_Subversion:**Size: 2 octets**

Value	Parameter Description
0XXXXX	Subversion of the Current LMP in the Controller. This value is vendor-specific.

Event(s) generated (unless masked away):

When the HCI_Read_Local_Version_Information command has completed, an HCI_Command_Complete event shall be generated.



7.4.2 Read Local Supported Commands command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Local_Supported_Commands	0x0002		Status, Supported_Commands

Description:

This command reads the list of HCI commands supported for the local Controller.

This command shall return the Supported_Commands configuration parameter.

See [Section 6.27](#) for more information.

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0	HCI_Read_Local_Supported_Commands command succeeded
0x01 to 0xFF	HCI_Read_Local_Supported_Commands command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Supported_Commands:

Size: 64 octets

Value	Parameter Description
	Bit mask for each HCI command. If a bit is 1, the Controller supports the corresponding command and the features required for the command. Unsupported or undefined commands shall be set to 0. See Section 6.27 .

Event(s) generated (unless masked away):

When the HCI_Read_Local_Supported_Commands command has completed, an HCI_Command_Complete event shall be generated.



7.4.3 Read Local Supported Features command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Local_Supported_Features	0x0003		Status, LMP_Features

Description:

This command requests a list of the supported features for the local BR/EDR Controller. This command will return a list of the LMP features. For details see [\[Vol 2\] Part C, Link Manager Protocol Specification](#).

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Local_Supported_Features command succeeded.
0x01 to 0xFF	HCI_Read_Local_Supported_Features command failed. See [Vol 1] Part F, Controller Error Codes .

LMP_Features:

Size: 8 octets

Value	Parameter Description
0xFFFFFFFF XXXXXXXX	Bit Mask List of LMP features. For details see [Vol 2] Part C, Link Manager Protocol Specification .

Event(s) generated (unless masked away):

When the HCI_Read_Local_Supported_Features command has completed, an HCI_Command_Complete event shall be generated.



7.4.4 Read Local Extended Features command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Local_Extended_Features	0x0004	Page_Number	Status, Page_Number, Max_Page_Number, Extended_LMP_Features

Description:

The HCI_Read_Local_Extended_Features command returns the requested page of the extended LMP features.

Command parameters:

Page_Number:

Size: 1 octet

Value	Parameter Description
0x00	Requests the normal LMP features as returned by Read_Local_Supported_Features.
0x01 to 0xFF	Return the corresponding page of features.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Local_Extended_Features command succeeded
0x01 to 0xFF	HCI_Read_Local_Extended_Features command failed. See [Vol 1] Part F, Controller Error Codes for list of error codes.

Page_Number:

Size: 1 octet

Value	Parameter Description
0x00	The normal LMP features as returned by Read_Local_Supported_Features.
0x01 to 0xFF	The page number of the features returned.

Max_Page_Number:

Size: 1 octet

Value	Parameter Description
0x00 to 0xFF	The highest features page number which contains non-zero bits for the local device.



Extended_LMP_Features:

Size: 8 octets

Value	Parameter Description
0xFFFFFFFFFFFFFFFF	Bit map of requested page of LMP features. See LMP specification for details.

Event(s) generated (unless masked away):

When the HCI_Read_Local_Extended_Features command has completed, an HCI_Command_Complete event shall be generated.



7.4.5 Read Buffer Size command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Buffer_Size	0x0005		Status, ACL_Data_Packet_Length, Synchronous_Data_Packet_Length, Total_Num_ACL_Data_Packets, Total_Num_Synchronous_Data_Packets

Description:

The HCI_Read_Buffer_Size command is used to read the maximum size of the data portion of HCI ACL and Synchronous Data packets sent from the Host to the Controller. The Host will segment the data to be transmitted from the Host to the Controller according to these sizes, so that the HCI Data packets will contain data with up to these sizes. The HCI_Read_Buffer_Size command also returns the total number of HCI ACL and Synchronous Data packets that can be stored in the data buffers of the Controller. The HCI_Read_Buffer_Size command shall be issued by the Host before it sends any data to the Controller.

For a device supporting BR/EDR and LE, if the HCI_LE_Read_Buffer_Size command returned zero for the number of buffers, then buffers returned by Read_Buffer_Size are shared between BR/EDR and LE.

The ACL_Data_Packet_Length return parameter will be used to determine the size of the L2CAP segments contained in ACL Data packets, which are transferred from the Host to the Controller to be broken up into Baseband packets by the Link Manager. The Synchronous_Data_Packet_Length return parameter is used to determine the maximum size of HCI Synchronous Data packets. The Total_Num_ACL_Data_Packets return parameter contains the total number of HCI ACL Data packets that can be stored in the data buffers of the Controller. The Host will determine how the buffers are to be divided between different Connection_Handles. The Total_Num_Synchronous_Data_Packets return parameter gives the same information but for HCI Synchronous Data packets.

The ACL_Data_Packet_Length and Synchronous_Data_Packet_Length return parameters do not include the length of the HCI ACL Data packet header or the HCI Synchronous Data packet header respectively.

Command parameters:

None.

**Return parameters:***Status:**Size: 1 octet*

Value	Parameter Description
0x00	HCI_Read_Buffer_Size command succeeded.
0x01 to 0xFF	HCI_Read_Buffer_Size command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

*ACL_Data_Packet_Length:**Size: 2 octets*

Value	Parameter Description
0xFFFF	Maximum length (in octets) of the data portion of each HCI ACL Data packet that the Controller is able to accept. Range: 0x0001 to 0xFFFF

*Synchronous_Data_Packet_Length:**Size: 1 octet*

Value	Parameter Description
0xFF	Maximum length (in octets) of the data portion of each HCI Synchronous Data packet that the Controller is able to accept.

*Total_Num_ACL_Data_Packets:**Size: 2 octets*

Value	Parameter Description
0xFFFF	Total number of HCI ACL Data packets that can be stored in the data buffers of the Controller. Range: 0x0001 to 0xFFFF

*Total_Num_Synchronous_Data_Packets:**Size: 2 octets*

Value	Parameter Description
0xFFFF	Total number of HCI Synchronous Data packets that can be stored in the data buffers of the Controller.

Event(s) generated (unless masked away):

When the HCI_Read_Buffer_Size command has completed, an HCI_Command_Complete event shall be generated.



7.4.6 Read BD_ADDR command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_BD_ADDR	0x0009		Status, BD_ADDR

Description:

On a BR/EDR Controller, this command reads the Bluetooth Controller address (BD_ADDR). (See [\[Vol 2\] Part B, Section 1.2](#) and [\[Vol 3\] Part C, Section 3.2.1](#)).

On an LE Controller, this command shall read the Public Device Address as defined in [\[Vol 6\] Part B, Section 1.3](#). If this Controller does not have a Public Device Address, the value 0x000000000000 shall be returned.

On a BR/EDR/LE Controller, the public address shall be the same as the BD_ADDR.

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_BD_ADDR command succeeded.
0x01 to 0xFF	HCI_Read_BD_ADDR command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0xFFFFFFFFXXXX	BD_ADDR of the Device

Event(s) generated (unless masked away):

When the HCI_Read_BD_ADDR command has completed, an HCI_Command_Complete event shall be generated.



7.4.7 Read Data Block Size command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Data_-_Block_Size	0x000A		Status, Max_ACL_Data_Packet_Length, Data_Block_Length, Total_Num_Data_Blocks

Description:

The HCI_Read_Data_Block_Size command is used to read values regarding the maximum permitted data transfers over the Controller and the data buffering available in the Controller.

The Host uses this information when fragmenting data for transmission, and when performing block-based flow control, based on the HCI_Number_Of_-_Completed_Data_Blocks event. The HCI_Read_Data_Block_Size command shall be issued by the Host before it sends any data to the Controller.

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Data_Block_Size command succeeded.
0x01 to 0xFF	HCI_Read_Data_Block_Size command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Max_ACL_Data_Packet_Length:

Size: 2 octets

Value	Parameter Description
0xFFFF	Maximum length (in octets) of the data portion of an HCI ACL Data packet that the Controller is able to accept for transmission.

Data_Block_Length:

Size: 2 octets

Value	Parameter Description
0xFFFF	Maximum length (in octets) of the data portion of each HCI ACL Data packet that the Controller is able to hold in each of its data block buffers.

**Total_Num_Data_Blocks:****Size: 2 octets**

Value	Parameter Description
0xFFFF	Total number of data block buffers available in the Controller for the storage of data packets scheduled for transmission.

Event(s) generated (unless masked away):

When the HCI_Read_Data_Block_Size command has completed, an HCI_Command_Complete event shall be generated.



7.4.8 Read Local Supported Codecs command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Local_Supported_Codecs [v2]	0x000D		Status, Num_Supported_Standard_Codecs, Standard_Codec_ID[i], Standard_Codec_Transport[i], Num_Supported_Vendor_Specific_Codecs, Vendor_Specific_Codec_ID[k], Vendor_Specific_Codec_Transport[k]
HCI_Read_Local_Supported_Codecs [v1]	0x000B		Status, Num_Supported_Standard_Codecs, Standard_Codec_ID[i], Num_Supported_Vendor_Specific_Codecs, Vendor_Specific_Codec_ID[k]

The order of the return parameters in an HCI event packet is:

```

Status
Num_Supported_Standard_Codecs
Standard_Codec_ID[0]
Standard_Codec_Transport[0]
...
Standard_Codec_ID[m]
Standard_Codec_Transport[m]
Num_Supported_Vendor_Specific_Codecs
Vendor_Specific_Codec_ID[0]
Vendor_Specific_Codec_Transport[0]
...
Vendor_Specific_Codec_ID[n]
Vendor_Specific_Codec_Transport[n]

```

Description:

This command reads a list of the Bluetooth SIG approved codecs supported by the Controller, as well as vendor specific codecs, which are defined by an individual manufacturer. The [v1] version of this command shall only return



codecs supported on the BR/EDR physical transport, while the [v2] version shall return codecs supported on all physical transports.

Command parameters:

None

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Local_Supported_Codecs command succeeded.
0x01 to 0xFF	HCI_Read_Local_Supported_Codecs command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Num_Supported_Standard_Codecs:

Size: 1 octet

Value	Parameter Description
0xXX	Total number of codecs supported

Standard_Codec_ID[i]:

Size: Num_Supported_Standard_Codecs × 1 octet

Value	Parameter Description
0xXX	Codec identifier. See Assigned Numbers

Standard_Codec_Transport[i]:

Size: Num_Supported_Standard_Codecs × 1 octets

Bit Number	Parameter Description
0	Codec supported over BR/EDR ACL
1	Codec supported over BR/EDR SCO and eSCO
2	Codec supported over LE CIS
3	Codec supported over LE BIS
All other bits	Reserved for future use

Num_Supported_Vendor_Specific_Codecs:

Size: 1 octet

Value	Parameter Description
0xXX	Total number of vendor-specific codecs supported

**Vendor_Specific_Codec_ID[k]:****Size:** *Num_Supported_Vendor_Specific_Codecs* × 4 octets

Value	Parameter Description
Octets 0 and 1	Company ID, see Assigned Numbers for Company Identifier
Octets 2 and 3	Vendor-defined codec ID

Vendor_Specific_Codec_Transport[k]:**Size:** *Num_Supported_Vendor_Specific_Codecs* × 1 octets

Bit Number	Parameter Description
0	Codec supported over BR/EDR ACL
1	Codec supported over BR/EDR SCO and eSCO
2	Codec supported over LE CIS
3	Codec supported over LE BIS
All other bits	Reserved for future use

Event(s) generated (unless masked away):

When the HCI_Read_Local_Supported_Codecs command has completed, an HCI_Command_Complete event shall be generated.



7.4.9 Read Local Simple Pairing Options command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Local_Simple_Pairing_Options	0x000C		Status, Simple_Pairing_Options, Max_Encryption_Key_Size

Description:

The HCI_Read_Local_Simple_Pairing_Options command is used to read the Secure Simple Pairing options and the maximum encryption key size supported. Bit 0 of the Simple_Pairing_Options return parameter shall be set to 1.

Note: If this command is supported, then the Controller must support remote public key validation (see [\[Vol 2\] Part H, Section 7.6](#)).

Command parameters:

None

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Local_Simple_Pairing_Options command succeeded.
0x01 to 0xFF	HCI_Read_Local_Simple_Pairing_Options command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Simple_Pairing_Options:

Size: 1 octet

Bit Number	Parameter Description
0	Remote public key validation is always performed.
All other bits	Reserved for future use.

Max_Encryption_Key_Size:

Size: 1 octet

Value	Parameter Description
0x07 to 0x10	Maximum encryption key size (in octets) supported.
All other values	Reserved for future use.

**Event(s) generated (unless masked away):**

When the HCI_Read_Local_Simple_Pairing_Options command has completed, an HCI_Command_Complete event shall be generated.



7.4.10 Read Local Supported Codec Capabilities command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Local_Supported_Codec_Capabilities	0x000E	Codec_ID, Logical_Transport_Type, Direction	Status, Num_Codec_Capabilities, Codec_Capability_Length[i], Codec_Capability[i]

Description:

The HCI_Read_Local_Supported_Codec_Capabilities command returns a list of codec capabilities supported by the Controller for a given codec. Only capabilities for the codec specified by the Codec_ID parameter and that match the transport specified by the Logical_Transport_Type parameter and direction specified by the Direction parameter are returned.

Note: The Controller cannot provide more information than will fit in an HCI Event packet. If more capabilities than that are available, it must select which ones to return. How this is done is not specified.

Command parameters:

Codec_ID:

Size: 5 octets

Value	Parameter Description
Octet 0	See Assigned Numbers
Octets 1 to 2	Company ID, see Assigned Numbers for Company Identifier. Shall be ignored if octet 0 is not 0xFF.
Octets 3 to 4	Vendor-defined codec ID. Shall be ignored if octet 0 is not 0xFF.

Logical_Transport_Type:

Size: 1 octet

Value	Parameter Description
0x00	BR/EDR ACL
0x01	BR/EDR SCO or eSCO
0x02	LE CIS
0x03	LE BIS
All other values	Reserved for future use

**Direction:****Size: 1 octet**

Value	Parameter Description
0x00	Input (Host to Controller)
0x01	Output (Controller to Host)
All other values	Reserved for future use

Return parameters:**Status:****Size: 1 octet**

Value	Parameter Description
0x00	HCI_Read_Local_Supported_Codec_Capabilities command succeeded
0x01 to 0xFF	HCI_Read_Local_Supported_Codec_Capabilities command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions

Num_Codec_Capabilities:**Size: 1 octet**

Value	Parameter Description
0xXX	Total number of codec capabilities returned

Codec_Capability_Length[i]:**Size: Num_Codec_Capabilities × 1 octets**

Value	Parameter Description
0xXX	Length of the Codec_Capability[i] field

Codec_Capability[i]:**Size: SUM(Codec_Capability_Length[i]) octets**

Value	Parameter Description
Variable	Codec_Capability_Length[i] octets of codec-specific capability data Note: Each element of this array has a variable length.

Event(s) generated (unless masked away):

When the HCI_Read_Local_Supported_Codec_Capabilities command has completed, an HCI_Command_Complete event shall be generated.



7.4.11 Read Local Supported Controller Delay command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Local_Supported_Controller_Delay	0x000F	Codec_ID, Logical_Transport_Type, Direction, Codec_Configuration_Length, Codec_Configuration	Status, Min_Controller_Delay, Max_Controller_Delay

Description:

The HCI_Read_Local_Supported_Controller_Delay command returns the range of supported Controller delays for the codec specified by the Codec_ID parameter on a given transport type specified by the Logical_Transport_Type parameter, in the direction specified by the Direction parameter, and with the codec configuration specified by the Codec_Configuration parameter.

The Min_Controller_Delay and Max_Controller_Delay parameters returned by the Controller provide a range of allowed values to be used by the Host when issuing the HCI_LE_Setup_ISO_Data_Path command.

The Min_Controller_Delay parameter returned by the Controller shall be greater than or equal to the codec processing delay for the specified direction and codec configuration.

The Max_Controller_Delay parameter returned by the Controller shall be less than or equal to the sum of the codec processing delay and the maximum time the Controller can buffer the data for the specified direction and codec configuration. Max_Controller_Delay shall be greater than or equal to Min_Controller_Delay.

Command parameters:

Codec_ID:

Size: 5 octets

Value	Parameter Description
Octet 0	See Assigned Numbers for Coding Format
Octets 1 to 2	Company ID, see Assigned Numbers for Company Identifier. Shall be ignored if octet 0 is not 0xFF.
Octets 3 to 4	Vendor-defined codec ID. Shall be ignored if octet 0 is not 0xFF.

*Logical_Transport_Type:**Size: 1 octet*

Value	Parameter Description
0x00	BR/EDR ACL
0x01	BR/EDR SCO or eSCO
0x02	LE CIS
0x03	LE BIS
All other values	Reserved for future use

*Direction:**Size: 1 octet*

Value	Parameter Description
0x00	Input (Host to Controller)
0x01	Output (Controller to Host)
All other values	Reserved for future use

*Codec_Configuration_Length:**Size: 1 octet*

Value	Parameter Description
0xXX	Length of codec configuration

*Codec_Configuration:**Size: Codec_Configuration_Length octets*

Value	Parameter Description
Variable	Codec-specific configuration data

Return parameters:*Status:**Size: 1 octet*

Value	Parameter Description
0x00	HCI_Read_Local_Supported_Controller_Delay command succeeded
0x01 to 0xFF	HCI_Read_Local_Supported_Controller_Delay command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

*Min_Controller_Delay:**Size: 3 octets*

Value	Parameter Description
0xXXXXXX	Minimum Controller delay in microseconds for the specified configuration Range: 0x000000 to 0x3D0900 Time range: 0 s to 4 s

**Max_Controller_Delay:****Size: 3 octets**

Value	Parameter Description
0xFFFFFFFF	Maximum Controller delay in microseconds for the specified configuration Range: 0x000000 to 0x3D0900 Time range: 0 s to 4 s

Event(s) generated (unless masked away):

When the HCI_Read_Local_Supported_Controller_Delay command has completed, an HCI_Command_Complete event shall be generated.



7.5 STATUS PARAMETERS

The Controller modifies all status parameters. These parameters provide information about the current state of the Link Manager and Baseband in the BR/EDR Controller. The Host device cannot modify any of these parameters other than to reset certain specific parameters.

For the status parameters commands, the OGF is defined as 0x05.

7.5.1 Read Failed Contact Counter command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Failed_Contact_Counter	0x0001	Handle	Status, Handle, Failed_Contact_Counter

Description:

This command reads the value for the Failed_Contact_Counter parameter for a particular connection to another device. The Handle shall be a Connection_Handle for an ACL connection. See [Section 6.15](#).

Command parameters:

Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xFFFF	The Handle for the Connection for which the Failed Contact Counter should be read. Range: 0x0000 to 0x0EFF

Return parameters:

Status: *Size: 1 octet*

Value	Parameter Description
0x00	HCI_Read_Failed_Contact_Counter command succeeded.
0x01 to 0xFF	HCI_Read_Failed_Contact_Counter command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xFFFF	The Handle for the connection for which the Failed Contact Counter has been read. Range: 0x0000 to 0x0EFF

**Failed_Contact_Counter:****Size: 2 octets**

Value	Parameter Description
0xFFFF	Number of consecutive failed contacts for a connection corresponding to the Handle.

Event(s) generated (unless masked away):

When the HCI_Read_Failed_Contact_Counter command has completed, an HCI_Command_Complete event shall be generated.



7.5.2 Reset Failed Contact Counter command

Command	OCF	Command Parameters	Return Parameters
HCI_Reset_Failed_Contact_Counter	0x0002	Handle	Status, Handle

Description:

This command resets the value for the Failed_Contact_Counter parameter for a particular connection to another device. The Handle shall be a Connection_Handle for an ACL connection. See [Section 6.15](#).

Command parameters:

Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xFFFF	The Handle for the connection for which the Failed Contact Counter should be reset. Range: 0x0000 to 0x0EFF

Return parameters:

Status: *Size: 1 octet*

Value	Parameter Description
0x00	HCI_Reset_Failed_Contact_Counter command succeeded.
0x01 to 0xFF	HCI_Reset_Failed_Contact_Counter command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xFFFF	The Handle for the connection for which the Failed Contact Counter has been reset. Range: 0x0000 to 0x0EFF

Event(s) generated (unless masked away):

When the HCI_Reset_Failed_Contact_Counter command has completed, an HCI_Command_Complete event shall be generated.



7.5.3 Read Link Quality command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Link_Quality	0x0003	Handle	Status, Handle, Link_Quality

Description:

This command returns the value for the Link_Quality for the specified Handle. The Handle shall be a Connection_Handle for an ACL connection. This command shall return a Link_Quality value from 0 to 255, which represents the quality of the link between two Controllers. The higher the value, the better the link quality is. Each Bluetooth module vendor will determine how to measure the link quality.

If the Host specifies a connection handle for an LE ACL connection and the Controller does not support the Connected Isochronous Stream - Central or Connected Isochronous Stream - Peripheral feature (see [\[Vol 6\] Part B, Section 4.6](#)), the Controller shall either begin to execute the command or return an error.

Command parameters:

Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xXXXX	The Handle for the connection for which link quality parameters are to be read. Range: 0x0000 to 0x0EFF

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Link_Quality command succeeded.
0x01 to 0xFF	HCI_Read_Link_Quality command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

**Handle:****Size: 2 octets (12 bits meaningful)**

Value	Parameter Description
0xXXXX	The Handle for the connection for which the link quality parameter has been read. Range: 0x0000 to 0x0EFF

Link_Quality:**Size: 1 octet**

Value	Parameter Description
0xXX	The current quality of the Link connection between the local device and the remote device specified by the Handle. Range: 0x00 to 0xFF The higher the value, the better the link quality is.

Event(s) generated (unless masked away):

When the HCI_Read_Link_Quality command has completed, an HCI_Command_Complete event shall be generated.



7.5.4 Read RSSI command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_RSSI	0x0005	Handle	Status, Handle, RSSI

Description:

This command reads the Received Signal Strength Indication (RSSI) value from a Controller.

For a BR/EDR Controller, the RSSI parameter returns the difference between the measured Received Signal Strength Indication (RSSI) and the limits of the Golden Receive Power Range for a Connection_Handle to another BR/EDR Controller. Any positive RSSI value returned by the Controller indicates how many dB the RSSI is above the upper limit, any negative value indicates how many dB the RSSI is below the lower limit. The value zero indicates that the RSSI is inside the Golden Receive Power Range.

Note: How accurate the dB values will be depends on the Bluetooth hardware. The only requirements for the hardware are that the BR/EDR Controller is able to tell whether the RSSI is inside, above or below the Golden Device Power Range.

The RSSI measurement compares the received signal power with two threshold levels, which define the Golden Receive Power Range. The lower threshold level corresponds to a received power between -56 dBm and 6 dB above the actual sensitivity of the receiver. The upper threshold level is 20 dB above the lower threshold level to an accuracy of ± 6 dB.

For an LE transport, the RSSI parameter returns the absolute receiver signal strength value in dBm to ± 6 dB accuracy. If the RSSI cannot be read, the RSSI metric shall be set to 127.

**Command parameters:***Handle:**Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xXXXX	The Handle for the connection for which the RSSI is to be read. The Handle is a Connection_Handle for an ACL-U or LE-U connection. Range: 0x0000 to 0x0EFF

Return parameters:*Status:**Size: 1 octet*

Value	Parameter Description
0x00	HCI_Read_RSSI command succeeded.
0x01 to 0xFF	HCI_Read_RSSI command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

*Handle:**Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xXXXX	The Handle for the connection for which the RSSI has been read. Range: 0x0000 to 0x0EFF

*RSSI:**Size: 1 octet*

Value	Parameter Description
0xFF	BR/EDR Range: -128 to 127 Units: dB LE: Range: -127 to 20, 127 Units: dBm

Event(s) generated (unless masked away):

When the HCI_Read_RSSI command has completed, an HCI_Command_-Complete event shall be generated.



7.5.5 Read AFH Channel Map command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_AFH_Channel_Map	0x0006	Connection_Handle	Status, Connection_Handle, AFH_Mode, AFH_Channel_Map

Description:

This command returns the values for the AFH_Mode and AFH_Channel_Map for the specified Connection_Handle. The Connection_Handle shall be a Connection_Handle for an ACL connection.

The returned values indicate the state of the hop sequence specified by the most recent LMP_SET_AFH message for the specified Connection_Handle, regardless of whether the Central has received the Baseband acknowledgment or whether the AFH_Instant has passed.

Command parameters:

Connection_Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xXXXX	Connection_Handle Range: 0x0000 to 0x0EFF

Return parameters:

Status: *Size: 1 octet*

Value	Parameter Description
0x00	HCI_Read_AFH_Channel_Map command succeeded.
0x01 to 0xFF	HCI_Read_AFH_Channel_Map command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Connection_Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xXXXX	Connection_Handle Range: 0x0000 to 0x0EFF

**AFH_Mode:****Size: 1 octet**

Value	Parameter Description
0x00	AFH disabled.
0x01	AFH enabled.
All other values	Reserved for future use.

AFH_Channel_Map:**Size: 10 octets (79 bits meaningful)**

Value	Parameter Description
0XXXXXXXXX XXXXXXXXXX XXX	<p>If AFH_Mode is not AFH enabled then the contents of this parameter are reserved for future use. Otherwise:</p> <p>This parameter contains 80 1-bit fields.</p> <p>The n^{th} such field (in the range 0 to 78) contains the value for channel n:</p> <p>0: channel n is unused</p> <p>1: channel n is used</p> <p>The most significant bit (bit 79) is reserved for future use</p>

Event(s) generated (unless masked away):

When the HCI_Read_AFH_Channel_Map command has completed, an HCI_Command_Complete event shall be generated.



7.5.6 Read Clock command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Clock	0x0007	Connection_Handle, Which_Clock	Status, Connection_Handle, Clock, Accuracy

Description:

This command reads the estimate of the value of the Bluetooth Clock from the BR/EDR Controller.

If the Which_Clock value is 0, then the Connection_Handle shall be ignored, the local Bluetooth Clock value shall be returned and the accuracy parameter shall be set to 0.

If the Which_Clock value is 1, then the Connection_Handle shall be a valid ACL Connection_Handle. If the current role of this ACL connection is Central, then the Bluetooth Clock of this device shall be returned. If the current role is Peripheral, then an estimate of the Bluetooth Clock of the remote Central and the accuracy of this value shall be returned.

The accuracy reflects the clock drift that might have occurred since the Peripheral last received a valid transmission from the Central.

Note: The Bluetooth Clock has a minimum accuracy of 250 ppm, or about 22 seconds drift in one day.

Note: See [\[Vol 2\] Part B, Section 1.1](#) for more information about the Bluetooth Clock.

Command parameters:

Connection_Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xXXXX	Connection_Handle Range: 0x0000 to 0x0EFF

Which_Clock: *Size 1 octet*

Value	Parameter Description
0xXX	0x00 = Local Clock (Connection_Handle does not have to be valid) 0x01 = Piconet Clock (Connection_Handle shall be valid) 0x02 to 0xFF = Reserved for future use

**Return parameters:***Status:**Size: 1 octet*

Value	Parameter Description
0x00	HCI_Read_Clock command succeeded.
0x01 to 0xFF	HCI_Read_Clock command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

*Connection_Handle:**Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xFFFF	The Connection_Handle for the connection for which the Central's clock has been read. If the Which_Clock parameter was 0, then the Connection_Handle is reserved for future use. Range: 0x0000 to 0x0EFF

*Clock:**Size: 4 octets (28 bits meaningful)*

Value	Parameter Description
0xFFFFFFFF	Bluetooth Clock of the device requested.

*Accuracy:**Size: 2 octets*

Value	Parameter Description
0xFFFF	Maximum (absolute) error in the Bluetooth Clock. Value of 0xFFFF means Unknown. Accuracy = $\pm N * 0.3125$ ms (1 Bluetooth Clock) Range: 0x0000 to 0xFFFE Time Range: 0 to 20479.375 ms

Event(s) generated (unless masked away):

When the HCI_Read_Clock command has completed, an HCI_Command_Complete event shall be generated.



7.5.7 Read Encryption Key Size command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Encryption_Key_Size	0x0008	Connection_Handle	Status, Connection_Handle, Key_Size

Description:

This command reads the current encryption key size associated with the Connection_Handle. The Connection_Handle shall be a Connection_Handle for an active ACL connection.

All BR/EDR Controllers shall implement this command.

Command parameters:

Connection_Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

Return parameters:

Status: *Size: 1 octet*

Value	Parameter Description
0x00	Read_Encryption_Key_Size succeeded
0x01 to 0xFF	Read_Encryption_Key_Size failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Connection_Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

Key_Size: *Size: 1 octet*

Value	Parameter Description
0xFF	Encryption key size. See [Vol 2] Part C, Section 5.2 .

**Event(s) generated (unless masked away):**

When the HCI_Read_Encryption_Key_Size command has completed, an HCI_Command_Complete event shall be generated.

If the ACL connection associated with the Connection_Handle is not encrypted, the Controller shall return an HCI_Command_Complete event with the error code *Insufficient Security* (0x2F).



7.5.8 [This section is no longer used]

7.5.9 [This section is no longer used]

7.5.10 [This section is no longer used]



7.5.11 Get MWS Transport Layer Configuration command

Command	OCF	Command Parameters	Return Parameters
HCI_Get_MWS_Transport_Layer_Configuration	0x000C		Status, Num_Transports, Transport_Layer[i], Num_Baud_Rates[i], To_MWS_Baud_Rate[k], From_MWS_Baud_Rate[k]

The order of the return parameters in an HCI event packet is:

```

Status
Num_Transports
Transport_Layer[0]
Num_Baud_Rates[0]
...
Transport_Layer[n]
Num_Baud_Rates[n]
To_MWS_Baud_Rate[0]
From_MWS_Baud_Rate[0]
...
To_MWS_Baud_Rate[m]
From_MWS_Baud_Rate[m]

```

Description:

The HCI_Get_MWS_Transport_Layer_Configuration command is used to inform the Host of the Baud rates supported by the Controller for the transport layer.

The Num_Transports parameter is used to indicate the number of MWS coexistence transport interfaces supported by the Controller.

The Num_Baud_Rates[i] parameter indicates the number of supported baud rates for each transport.

The To_MWS_Baud_Rate[k] parameters indicate the supported baud rates in the direction from Bluetooth to MWS for each transport.

The From_MWS_Baud_Rate[k] parameters indicate the supported baud rates in the direction from MWS to Bluetooth for each transport.



If one direction has more supported rates than the other direction, the Controller shall - in the direction with less supported rates - fill with sufficient zeros to produce the same number of values. The rates for the two directions are not necessarily paired.

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Get_MWS_Transport_Layer_Configuration command succeeded.
0x01 to 0xFF	HCI_Get_MWS_Transport_Layer_Configuration command failed. See [Vol 1] Part F, Controller Error Codes , for error codes and descriptions.

Num_Transports:

Size: 1 octet

Value	Parameter Description
0xXX	Number of supported MWS coexistence transport layers.

Transport_Layer[i]:

Size: Num_Transports × 1 octet

Value	Parameter Description
0xXX	See Assigned Numbers .

Num_Baud_Rates[i]:

Size: Num_Transports × 1 octet

Value	Parameter Description
0xXX	Number of different baud rates supported for one transport.

To_MWS_Baud_Rate[k]:

Size: SUM (Num_Baud_Rates [i]) × 4 octets

Value	Parameter Description
0XXXXXXXX	A supported Baud rate in the Bluetooth Controller to MWS Device direction in Baud. The list shall start with the first baud rate for the first transport, followed by the remaining baud rates for the first transport, followed by the baud rates for the second transport (if any), followed by baud rates for subsequent transports (if any).



From_MWS_Baud_Rate[k]: Size: SUM (Num_Baud_Rates[i]) × 4 octets

Value	Parameter Description
0xFFFFFFFF	A supported Baud rate in the Bluetooth Controller for signals in the MWS to Bluetooth Controller Device direction in Baud. The list shall start with the first baud rate for the first transport, followed by the remaining baud rates for the first transport, followed by the baud rates for the second transport (if any), followed by baud rates for subsequent transports (if any).

Event(s) generated (unless masked away):

When the HCI_Get_MWS_Transport_Layer_Configuration command has completed, an HCI_Command_Complete event shall be generated.



7.5.12 Set Triggered Clock Capture command

Command	OCF	Command Parameters	Return Parameters
HCI_Set_Triggered_Clock_Capture	0x000D	Connection_Handle, Enable, Which_Clock, LPO_Allowed, Num_Clock_Captures_ To_Filter	Status

Description:

The HCI_Set_Triggered_Clock_Capture command configures the BR/EDR Controller for triggered clock capturing.

Triggered clock capturing is enabled or disabled by the Enable parameter. If Enable is set to 0x00, all the other parameters (including Connection_Handle) shall be ignored.

If the Which_Clock value is 0, then the Connection_Handle shall be ignored. If the Which_Clock value is 1, then the Connection_Handle shall be a valid ACL Connection_Handle.

The LPO_Allowed parameter informs the BR/EDR Controller whether it may use a lower accuracy clock or not.

The Num_Clock_Captures_To_Filter parameter is used to filter triggered clock captures between sending HCI_Triggered_Clock_Capture events to the Host. When set to zero, all triggered clock captures shall result in an HCI_Triggered_Clock_Capture event sent to the Host. When set to a non-zero value, after every HCI_Triggered_Clock_Capture event, Num_Clock_Captures_To_Filter triggered clock captures in a row shall not trigger an event to be sent to the Host.

Note: An implementation should ensure that the rate of triggered clock captures does not overwhelm the HCI event queue and processing.

Note: See [\[Vol 2\] Part B, Section 1.1](#) for more information about the Bluetooth Clock.

Command parameters:

Connection_Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xXXXX	Connection_Handle Range: 0x0000 to 0x0EFF

**Enable:****Size: 1 octet**

Value	Parameter Description
0x00	Disable triggered clock capturing on the specified Connection_Handle (Default)
0x01	Enable triggered clock capturing on the specified Connection_Handle
All other values	Reserved for future use

Which_Clock:**Size: 1 octet**

Value	Parameter Description
0x00	Local Clock
0x01	Piconet Clock for the specified connection
All other values	Reserved for future use

LPO_Allowed:**Size: 1 octet**

Value	Parameter Description
0x00	Controller shall not sleep (that is, clock accuracy shall be equal to or better than ± 20 ppm)
0x01	Controller may sleep (that is, clock accuracy shall be equal to or better than ± 250 ppm)
All other values	Reserved for future use

Num_Clock_Captures_To_Filter:**Size: 1 octet**

Value	Parameter Description
0x00	All triggered clock captures result in an HCI_Triggered_Clock_Capture event sent to the Host
0x01 to 0xFF	Number of triggered clock captures filtered between sending an HCI_Triggered_Clock_Capture event to the Host.

Return parameters:**Status:****Size: 1 octet**

Value	Parameter Description
0x00	HCI_Set_Triggered_Clock_Capture command succeeded.
0x01 to 0xFF	HCI_Set_Triggered_Clock_Capture command failed. See [Vol 1] Part F, Controller Error Codes , for error codes and descriptions.

**Event(s) generated (unless masked away):**

When the HCI_Set_Triggered_Clock_Capture command has completed, an HCI_Command_Complete event shall be sent to the Host.

When Triggered Clock Capturing is enabled, HCI_Triggered_Clock_Capture events are returned until Triggered Clock Capturing is disabled.



7.6 TESTING COMMANDS

The Testing commands are used to provide the ability to test various functional capabilities of the Bluetooth hardware. These commands provide the ability to arrange various conditions for testing.

For the Testing commands, the OGF is defined as 0x06.

7.6.1 Read Loopback Mode command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Loopback_Mode	0x0001		Status, Loopback_Mode

Description:

This command reads the value for the setting of the Controller's Loopback_Mode. The setting of the Loopback_Mode parameter shall determine the path of information. In Non-testing Mode operation, the Loopback_Mode parameter is set to Non-testing Mode and the path of the information is as specified by the Bluetooth specifications. In Local Loopback Mode, every data packet (ACL, SCO and eSCO) and command packet that is sent from the Host to the Controller is sent back with no modifications by the Controller, as shown in [Figure 7.1](#). For details of loopback modes see [Section 7.6.2](#).

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Loopback_Mode command succeeded.
0x01 to 0xFF	HCI_Read_Loopback_Mode command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Loopback_Mode:

Size: 1 octet

Value	Parameter Description
0x00	No Loopback mode enabled (default).
0x01	Enable Local Loopback.
0x02	Enable Remote Loopback.
All other values	Reserved for future use.

**Event(s) generated (unless masked away):**

When the HCI_Read_Loopback_Mode command has completed, an HCI_Command_Complete event shall be generated.



7.6.2 Write Loopback Mode command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Loopback_Mode	0x0002	Loopback_Mode	Status

Description:

This command writes the value for the setting of the BR/EDR Controller’s Loopback mode. The setting of the Loopback_Mode parameter shall determine the path of information. In Non-testing mode operation, the Loopback_Mode parameter is set to Non-testing mode and the path of the information as specified by the Bluetooth specifications. In Local Loopback mode, every data packet (ACL, SCO and eSCO) and command packet that is sent from the Host to the BR/EDR Controller is sent back with no modifications by the BR/EDR Controller, as shown in [Figure 7.1](#).

When the BR/EDR Controller enters Local Loopback mode, it shall respond with one to four Connection_Handles, one for an ACL connection and zero to three for synchronous connections. The Host should use these Connection_Handles when sending data in Local Loopback mode. The number of Connection_Handles returned for synchronous connections (between zero and three) is implementation specific. When in Local Loopback mode, the BR/EDR Controller loops back commands and data to the Host. The HCI_Loopback_Command event is used to loop back commands that the Host sends to the Controller.

There are some commands that are not looped back in Local Loopback mode: HCI_Reset, HCI_Set_Controller_To_Host_Flow_Control, HCI_Host_Buffer_Size, HCI_Host_Number_Of_Completed_Packets, HCI_Read_Buffer_Size, HCI_Read_Loopback_Mode and HCI_Write_Loopback_Mode. These commands should be executed in the way they are normally executed. The commands HCI_Reset and HCI_Write_Loopback_Mode can be used to exit Local Loopback mode.

If HCI_Write_Loopback_Mode is used to exit Local Loopback mode on a BR/EDR Controller, HCI_Disconnection_Complete events corresponding to the HCI_Connection_Complete events that were sent when entering Local Loopback mode should be sent to the Host. Furthermore, no connections are allowed in Local Loopback mode. If there is a connection, and there is an attempt to set the device to Local Loopback mode, the attempt will be refused. When the device is in Local Loopback mode, the Controller will refuse incoming connection attempts. This allows the Host BR/EDR Controller Transport Layer to be tested without any other variables.

If a BR/EDR Controller is set to Remote Loopback mode, it will send back all data (ACL, SCO and eSCO) that comes over the air. It will only allow a maximum of one ACL connection and three synchronous connections, and these shall all be to the same remote device. If there are existing connections



to a remote device and there is an attempt to set the local device to Remote Loopback mode, the attempt shall be refused.

See [Figure 7.2](#), where the rightmost device is set to Remote Loopback mode and the leftmost device is set to Non-testing mode. This allows the BR/EDR Air link to be tested without any other variables.

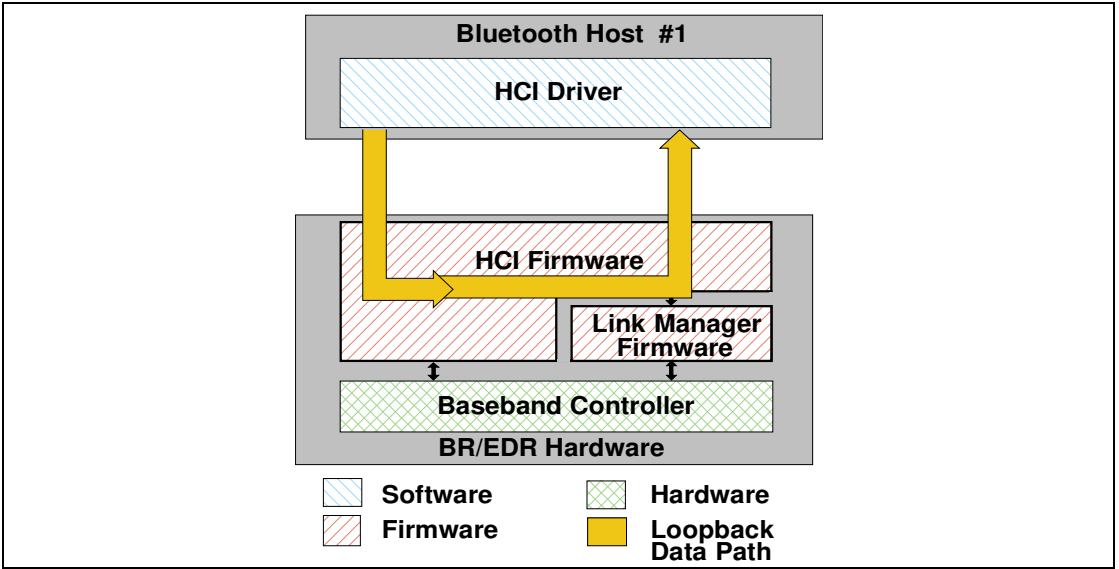


Figure 7.1: Local Loopback mode

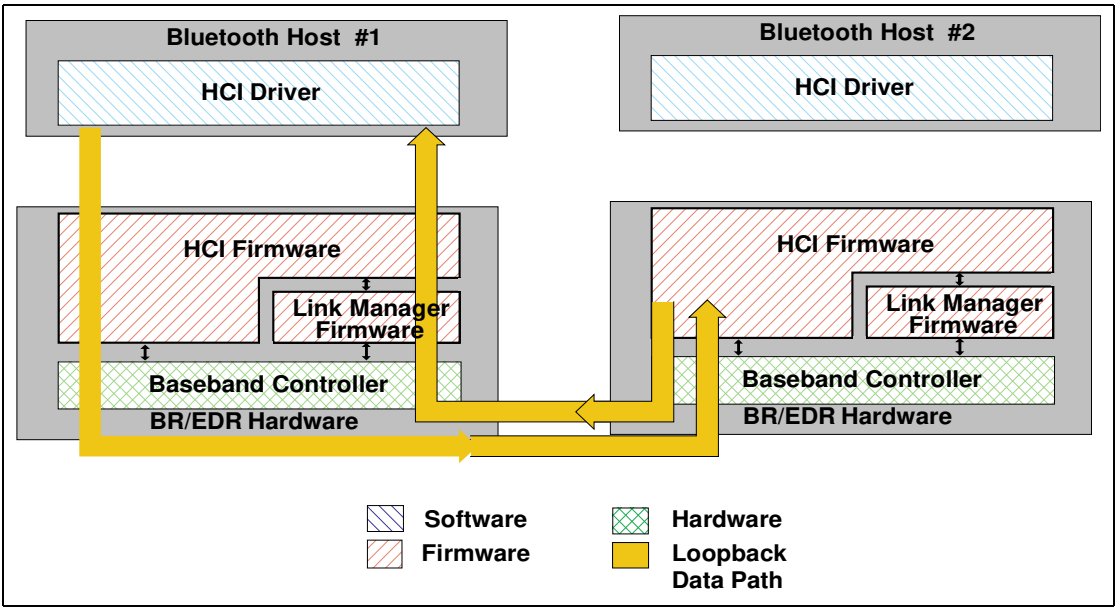


Figure 7.2: Remote Loopback mode

**Command parameters:***Loopback_Mode:**Size: 1 octet*

Value	Parameter Description
0x00	No Loopback mode enabled (default).
0x01	Enable Local Loopback.
0x02	Enable Remote Loopback.
All other values	Reserved for future use.

Return parameters:*Status:**Size: 1 octet*

Value	Parameter Description
0x00	HCI_Write_Loopback_Mode command succeeded.
0x01 to 0xFF	HCI_Write_Loopback_Mode command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

When the HCI_Write_Loopback_Mode command has completed, an HCI_Command_Complete event shall be generated.



7.6.3 Enable Device Under Test Mode command

Command	OCF	Command Parameters	Return Parameters
HCI_Enable_Device_Under_Test_Mode	0x0003		Status

Description:

The HCI_Enable_Device_Under_Test_Mode command allows the local BR/EDR Controller to enter test mode via LMP test commands for BR/EDR Controllers. For details see [\[Vol 2\] Part C, Link Manager Protocol Specification](#). The Host issues this command when it wants the local device to be the DUT for the Testing scenarios as described in [\[Vol 3\] Part D, Section 1](#). When the BR/EDR Controller receives this command, it shall complete the command with an HCI_Command_Complete event. The BR/EDR Controller functions as normal until the remote tester issues the LMP test command to place the local device into Device Under Test mode. To disable and exit the Device Under Test Mode, the Host may issue the HCI_Reset command. The local BR/EDR Controller shall not enter test mode, even if instructed by the remote BR/EDR Controller, before this command is issued.

Command parameters:

None.

Return parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Enter_Device_Under_Test_Mode command succeeded.
0x01 to 0xFF	HCI_Enter_Device_Under_Test_Mode command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

When the HCI_Enter_Device_Under_Test_Mode command has completed, an HCI_Command_Complete event shall be generated.



7.6.4 Write Simple Pairing Debug Mode command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Simple_Pairing_Debug_Mode	0x0004	Simple_Pairing_Debug_Mode	Status

Description:

This command configures the BR/EDR Controller to use a predefined Diffie Hellman private key for Secure Simple Pairing to enable debug equipment to monitor the encrypted connection.

Note: Only one side (initiator or responder) needs to set Secure Simple Pairing debug mode in order for debug equipment to be able to determine the link key and, therefore, be able to monitor the encrypted connection.

When the Simple_Pairing_Debug_Mode parameter is set to enabled the BR/EDR Controller shall use the predefined Diffie Hellman private key. The BR/EDR Controller shall also set the resulting Link_Key type to "Debug Combination Key."

When in Secure Simple Pairing debug mode, the Link Manager shall use the following Diffie Hellman private / public key pairs:

For P-192:

Private key: 07915f86918ddc27005df1d6cf0c142b625ed2eff4a518ff

Public key (X): 15207009984421a6586f9fc3fe7e4329d2809ea51125f8ed

Public key (Y): b09d42b81bc5bd009f79e4b59dbbaa857fca856fb9f7ea25

For P-256:

Private key: 3f49f6d4 a3c55f38 74c9b3e3 d2103f50 4aff607b eb40b799 5899b8a6 cd3c1abd

Public key (X): 20b003d2 f297be2c 5e2c83a7 e9f9a5b9 eff49111 acf4fddb cc030148 0e359de6

Public key (Y): dc809c49 652aeb6d 63329abf 5a52155c 766345c2 8fed3024 741c8ed0 1589d28b

Command parameters:

Simple_Pairing_Debug_Mode:

Size: 1 octet

Value	Parameter Description
0x00	Secure Simple Pairing debug mode disabled (default)
0x01	Secure Simple Pairing debug mode enabled
All other values	Reserved for future use

**Return parameters:***Status:**Size: 1 octet*

Value	Parameter Description
0x00	HCI_Write_Simple_Pairing_Debug_Mode command succeeded.
0x01 to 0xFF	HCI_Write_Simple_Pairing_Debug_Mode command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Event(s) generated (unless masked away):

When the HCI_Write_Simple_Pairing_Debug_Mode command has completed, an HCI_Command_Complete event shall be generated.



7.6.5 [This section is no longer used]

7.6.6 [This section is no longer used]

7.6.7 [This section is no longer used]



7.6.8 Write Secure Connections Test Mode command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Secure_- Connections_Test_Mode	0x000A	Connection_Handle, DM1_ACL-U_Mode, eSCO_Loopback_Mode	Status, Connection_Handle

Description:

This command configures the BR/EDR Controller to enable and disable the two test modes used for verifying the Secure Connections feature during qualification. The DM1_ACL-U_Mode parameter enables and disables the use of DM1 packets for transmitting ACL-U data. When DM1 ACL-U Mode is disabled, ACL-U traffic may use DM1 packets. When DM1 ACL-U Mode is enabled, ACL-U traffic shall not use DM1 packets unless the Packet_Type parameter only allows DM1 packets (e.g. set to 0x3306 or 0x330E).

The command is used during testing to help make transmit ACL packet selection predictable.

The eSCO_Loopback_Mode parameter enables and disables the loopback of received eSCO payloads. When the eSCO_Loopback_Mode parameter is set to Enabled, the BR/EDR Controller will send back all eSCO data that comes over the air irrespective of whether the CRC check in the received eSCO packet passes or fails. It will only allow one synchronous connection. If there is more than one synchronous connection and there is an attempt to set the local device to eSCO_Loopback_Mode, the attempt shall be refused.

See [Figure 7.3](#), where the rightmost device has the eSCO_Loopback_Mode parameter set to enabled and the leftmost device is in a normal mode of operation. This allows the encryption and decryption of eSCO packets to be tested without any other variables.

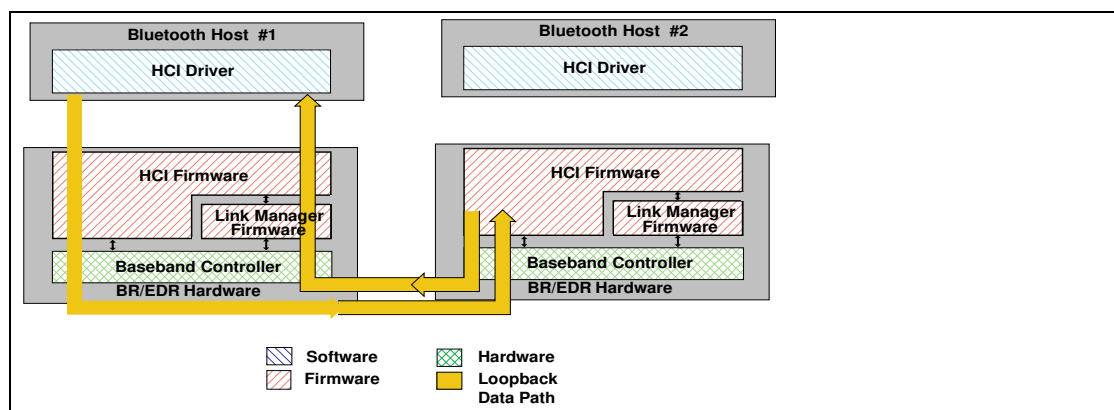


Figure 7.3: Secure Connections eSCO Loopback

The Connection_Handle shall be for an ACL connection.

When the eSCO_Loopback_Mode parameter is set to enabled, received eSCO payloads are looped back as subsequent transmitted eSCO payloads. There may be a delay of 0 or more eSCO intervals before the Controller loops back the payload. This is illustrated in [Figure 7.4](#), [Figure 7.5](#), and [Figure 7.6](#).

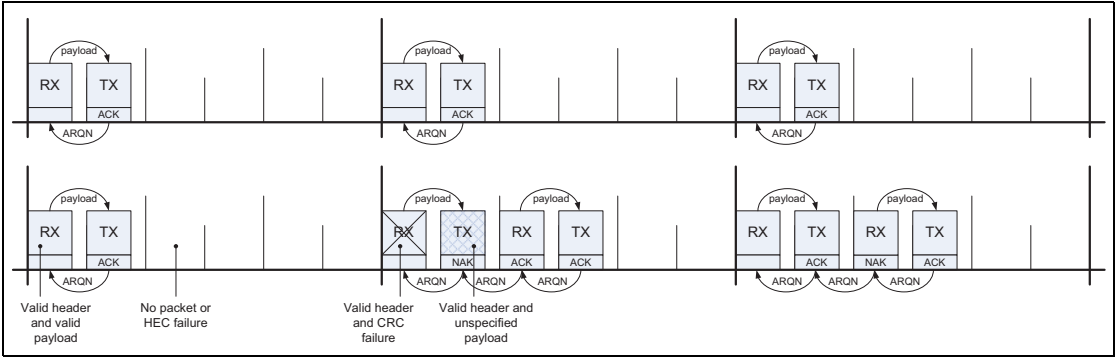


Figure 7.4: Secure Connections eSCO loopback immediate

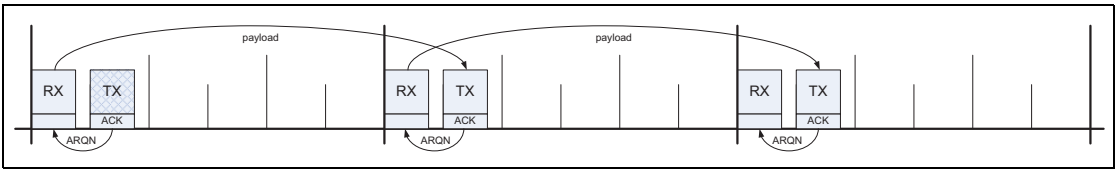


Figure 7.5: Secure Connections eSCO loopback delayed

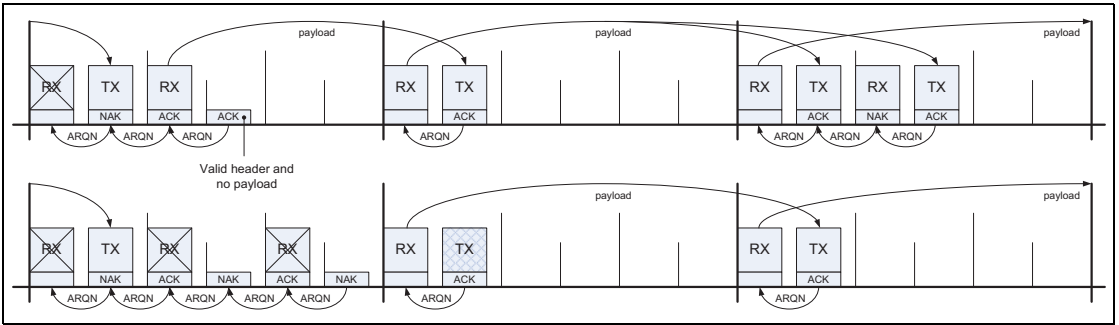


Figure 7.6: Secure Connections eSCO loopback delayed with retransmissions

Command parameters:

Connection_Handle: Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xXXXX	Connection_Handle Range: 0x0000 to 0x0EFF

**DM1_ACL-U_Mode:****Size: 1 octet**

Value	Parameter Description
0x00	DM1 ACL-U mode disabled (default)
0x01	DM1 ACL-U mode enabled
All other values	Reserved for future use

eSCO_Loopback_Mode:**Size: 1 octet**

Value	Parameter Description
0x00	eSCO loopback mode disabled (default)
0x01	eSCO loopback mode enabled
All other values	Reserved for future use

Return parameters:**Status:****Size: 1 octet**

Value	Parameter Description
0x00	HCI_Write_Secure_Connections_Test_Mode command succeeded.
0x01 to 0xFF	HCI_Write_Secure_Connections_Test_Mode command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Connection_Handle:**Size: 2 octets (12 bits meaningful)**

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

Event(s) generated (unless masked away):

When the HCI_Write_Secure_Connections_Test_Mode command has completed, an HCI_Command_Complete event shall be generated.



7.7 EVENTS

7.7.1 Inquiry Complete event

Event	Event Code	Event Parameters
HCI_Inquiry_Complete	0x01	Status

Description:

The HCI_Inquiry_Complete event indicates that the Inquiry is finished. This event contains a Status parameter, which is used to indicate if the Inquiry completed successfully or if the Inquiry was not completed.

Event parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Inquiry command completed successfully.
0x01 to 0xFF	HCI_Inquiry command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.



7.7.2 Inquiry Result event

Event	Event Code	Event Parameters
HCI_Inquiry_Result	0x02	Num_Responses, BD_ADDR[i], Page_Scan_Repetition_Mode[i], Reserved[i], Class_Of_Device[i] Clock_Offset[i]

Description:

The HCI_Inquiry_Result event indicates that a BR/EDR Controller or multiple BR/EDR Controllers have responded so far during the current Inquiry process. This event will be sent from the BR/EDR Controller to the Host as soon as an Inquiry Response from a remote device is received if the remote device supports only mandatory paging scheme. The BR/EDR Controller may queue these Inquiry Responses and send multiple BR/EDR Controllers information in one HCI_Inquiry_Result event. The event can be used to return one or more Inquiry responses in one event.

This event is only generated if the Inquiry_Mode parameter of the last HCI_Write_Inquiry_Mode command was set to 0x00 (Standard Inquiry Result event format) or if the HCI_Write_Inquiry_Mode command has not been used.

Event parameters:

Num_Responses: *Size: 1 octet*

Value	Parameter Description
0xXX	Number of responses from the Inquiry.

BD_ADDR[i]: *Size: Num_Responses × 6 octets*

Value	Parameter Description
0XXXXXXXXXX XX	BD_ADDR for a device which responded.

Page_Scan_Repetition_Mode[i]: *Size: Num_Responses × 1 octet*

Value	Parameter Description
0x00	R0
0x01	R1
0x02	R2
All other values	Reserved for future use

**Reserved[i]:****Size:** *Num_Responses* × 2 octets

Value	Parameter Description
0xFFFF	Reserved for future use.

Class_Of_Device[i]:**Size:** *Num_Responses* × 3 octets

Value	Parameter Description
0xFFFFF	Class of Device for the device

Clock_Offset[i]:**Size:** *Num_Responses* × 2 octets

Bit Number	Parameter Description
0-14	Bits 16-2 of CLKNPeripheral - CLK
15	Reserved for future use



7.7.3 Connection Complete event

Event	Event Code	Event Parameters
HCI_Connection_Complete	0x03	Status, Connection_Handle, BD_ADDR, Link_Type, Encryption_Enabled

Description:

The HCI_Connection_Complete event indicates to both of the Hosts forming the connection that a new connection has been established. This event also indicates to the Host which issued the HCI_Create_Connection, HCI_Accept_Connection_Request, or HCI_Reject_Connection_Request command, and then received an HCI_Command_Status event, if the issued command failed or was successful.

Event parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	Connection successfully completed.
0x01 to 0xFF	Connection failed to Complete. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Connection_Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xFFFF	Connection_Handle for the new connection Range: 0x0000 to 0x0EFF

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0XXXXXXXXXXXXX	BD_ADDR of the other connected Device forming the connection.

Link_Type:

Size: 1 octet

Value	Parameter Description
0x00	SCO connection.
0x01	ACL connection (Data Channels).
All other values	Reserved for future use.

Host Controller Interface Functional Specification***Encryption_Enabled:******Size: 1 octet***

Value	Parameter Description
0x00	Link level encryption disabled.
0x01	Link level encryption enabled.
All other values	Reserved for future use.



7.7.4 Connection Request event

Event	Event Code	Event Parameters
HCI_Connection_Request	0x04	BD_ADDR, Class_Of_Device, Link_Type

Description:

The HCI_Connection_Request event is used to indicate that a new incoming connection is trying to be established. The connection may either be accepted or rejected. When the Host receives this event and the link type parameter is ACL, it should respond with either an HCI_Accept_Connection_Request or HCI_Reject_Connection_Request command before the timer Connection_Accept_Timeout expires. If the link type is SCO or eSCO, the Host should reply with the HCI_Accept_Synchronous_Connection_Request command, the HCI_Enhanced_Accept_Synchronous_Connection_Request command, or the HCI_Reject_Synchronous_Connection_Request command. If the link type is SCO, the Host may respond with the HCI_Accept_Connection_Request command. If the event is responded to with the HCI_Accept_Connection_Request command, then the default parameter settings of the HCI_Accept_Synchronous_Connection_Request command (see [Section 7.1.27](#)) should be used by the local Link Manager when negotiating the SCO link parameters. In that case, the HCI_Connection_Complete event and not the HCI_Synchronous_Connection_Complete event, shall be returned on completion of the connection.

Note: See [Section 7.3.3](#) for the behavior when the HCI_Connection_Request event is masked or the connection is auto accepted.

Event parameters:

BD_ADDR: Size: 6 octets

Value	Parameter Description
0xFFFFFFFFXXXX	BD_ADDR of the device that requests the connection.

Class_Of_Device: Size: 3 octets

Value	Parameter Description
0xFFFFFFFF	Class of Device for the device, which requests the connection.
0x000000	Unknown Class of Device

Link_Type: Size: 1 octet

Value	Parameter Description
0x00	SCO connection requested
0x01	ACL connection requested



Value	Parameter Description
0x02	eSCO connection requested
All other values	Reserved for future use.



7.7.5 Disconnection Complete event

Event	Event Code	Event Parameters
HCI_Disconnection_Complete	0x05	Status, Connection_Handle, Reason

Description:

The HCI_Disconnection_Complete 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, the value of the reason parameter shall be ignored by the Host. For example, this can be the case if the Host has issued the HCI_Disconnect command and there was a parameter error, or the command was not presently allowed, or a Connection_Handle that didn't correspond to a connection was given.

If Connection_Handle identifies a CIS on the Central, the handle and the associated data paths of that CIS shall remain valid (irrespective of whether the disconnection was successful or not). If Connection_Handle identifies a CIS on the Peripheral and Status is zero, the handle and the associated data path of that CIS shall be deleted. If Connection_Handle identifies any other kind of connection and Status is zero, the handle shall be deleted.

Note: When a physical link fails, one HCI_Disconnection_Complete event will be returned for each logical channel on the physical link with the corresponding Connection_Handle as a parameter.

Event parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	Disconnection has occurred.
0x01 to 0xFF	Disconnection failed to complete. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Connection_Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xFFFF	Connection_Handle which was disconnected. Range: 0x0000 to 0xFFFF

Host Controller Interface Functional Specification*Reason:**Size: 1 octet*

Value	Parameter Description
0xXX	Reason for disconnection. See [Vol 1] Part F, Controller Error Codes for error codes and descriptions.



7.7.6 Authentication Complete event

Event	Event Code	Event Parameters
HCI_Authentication_Complete	0x06	Status, Connection_Handle

Description:

The HCI_Authentication_Complete event occurs when authentication has been completed for the specified connection. The Connection_Handle shall be a Connection_Handle for an ACL connection.

Note: This event is only generated on the initiator of the authentication and not on the responder.

Event parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	Authentication Request successfully completed.
0x01 to 0xFF	Authentication Request failed to complete. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Connection_Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF



7.7.7 Remote Name Request Complete event

Event	Event Code	Event Parameters
HCI_Remote_Name_Request_Complete	0x07	Status, BD_ADDR, Remote_Name

Description:

The HCI_Remote_Name_Request_Complete event is used to indicate that a remote name request has been completed. The Remote_Name event parameter is a UTF-8 encoded string with the type utf8{248} defined in [\[Vol 1\] Part E, Section 2.9.3](#). The BD_ADDR event parameter is used to identify which device the user-friendly name was obtained from.

Note: The Remote_Name parameter is a string parameter. Endianness does therefore not apply to the Remote_Name parameter. The first octet of the name is received first.

Event parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Remote_Name_Request command succeeded.
0x01 to 0xFF	HCI_Remote_Name_Request command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0XXXXXXXXXXXXX	BD_ADDR for the device whose name was requested.

Remote_Name:

Size: 248 octets

Value	Parameter Description
Name[248]	A UTF-8 encoded user-friendly descriptive name for the remote device with type utf8{248}.



7.7.8 Encryption Change event

Event	Event Code	Event Parameters
HCI_Encryption_Change [v2]	0x59	Status, Connection_Handle, Encryption_Enabled, Encryption_Key_Size
HCI_Encryption_Change [v1]	0x08	Status, Connection_Handle, Encryption_Enabled

Description:

The HCI_Encryption_Change event is used to indicate that the change of the encryption mode has been completed. The Connection_Handle event parameter will be a Connection_Handle for an ACL connection and is used to identify the remote device. The Encryption_Enabled event parameter specifies the new encryption state for the connection specified by Connection_Handle. The Encryption_Key_Size event parameter specifies the size, in octets, of the key used to encrypt the link. This event will occur on both devices to notify the Hosts when encryption has changed for all connections between the two devices. This event shall not be generated if encryption is paused or resumed; during a role switch, for example.

Event parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	An encryption change has occurred.
0x01 to 0xFF	An attempt to change encryption failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Connection_Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

Encryption_Enabled:

Size: 1 octet

Value	Parameter Description
0x00	Link Level Encryption is OFF.
0x01	Link Level Encryption is ON with E0 for BR/EDR. Link Level Encryption is ON with AES-CCM for LE.



Value	Parameter Description
0x02	Link Level Encryption is ON with AES-CCM for BR/EDR.
All other values	Reserved for future use.

*Encryption_Key_Size:**Size: 1 octet*

Value	Parameter Description
0xXX	Encryption key size in octets. This parameter shall be ignored for LE connections and shall be ignored when Link Level Encryption is OFF. Range: 0x01 to 0x10



7.7.9 Change Connection Link Key Complete event

Event	Event Code	Event Parameters
HCI_Change_Connection_Link_Key_ - Complete	0x09	Status, Connection_Handle

Description:

The HCI_Change_Connection_Link_Key_Complete event is used to indicate that the change in the Link Key for all connections to a given remote BR/EDR Controller has been completed.

The Connection_Handle will be a Connection_Handle for an ACL connection to the remote Controller. The HCI_Change_Connection_Link_Key_Complete event is sent only to the Host which issued the HCI_Change_Connection_Link_Key command.

Event parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Change_Connection_Link_Key command succeeded.
0x01 to 0xFF	HCI_Change_Connection_Link_Key command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Connection_Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF



7.7.10 Link Key Type Changed event

Event	Event Code	Event Parameters
HCI_Link_Key_Type_Changed	0x0A	Status, Connection_Handle, Key_Flag

Description:

The HCI_Link_Key_Type_Changed event is used to indicate that the Link Key managed by the Central of the piconet has been changed. The Connection_Handle will be a Connection_Handle for an ACL connection within that piconet. The link key used for the connection will be the temporary link key or the semi-permanent link key indicated by the Key_Flag. The Key_Flag event parameter is used to indicate which Link Key (temporary link key or the semi-permanent link keys) is now being used in the piconet.

This event is also generated on the local Controller when the HCI_Link_Key_Selection command finishes because no change was requested or an error occurred. If a key change was attempted but failed, the remote Controller may generate the event.

Note: For a Central, the change from a semi-permanent Link Key to temporary Link Key will affect all Connection_Handles related to the piconet. For a Peripheral, this change affects only this particular Connection_Handle. A temporary link key must be used when both broadcast and point-to-point traffic are being encrypted.

Event parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Link_Key_Selection command succeeded.
0x01 to 0xFF	HCI_Link_Key_Selection command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Connection_Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

**Key_Flag:****Size: 1 octet**

Value	Parameter Description
0x00	Using Semi-permanent Link Key.
0x01	Using Temporary Link Key.



7.7.11 Read Remote Supported Features Complete event

Event	Event Code	Event Parameters
HCI_Read_Remote_Supported_Features_Complete	0x0B	Status, Connection_Handle, LMP_Features

Description:

The HCI_Read_Remote_Supported_Features_Complete event is used to indicate the completion of the process of the Link Manager obtaining the supported features of the remote BR/EDR Controller specified by the Connection_Handle event parameter. The Connection_Handle will be a Connection_Handle for an ACL connection. The event parameters include a list of LMP features. For details see [\[Vol 2\] Part C, Link Manager Protocol Specification](#).

Note: If the features are requested more than once while a connection exists between the two devices, the second and subsequent requests may report a cached copy of the features rather than fetching the feature mask again.

Event parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Remote_Supported_Features command succeeded.
0x01 to 0xFF	HCI_Read_Remote_Supported_Features command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Connection_Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

LMP_Features:

Size: 8 octets

Value	Parameter Description
0xFFFFFFFF XXXXXXXX	Bit Mask List of LMP features. See [Vol 2] Part C, Link Manager Protocol Specification .



7.7.12 Read Remote Version Information Complete event

Event	Event Code	Event Parameters
HCI_Read_Remote_Version_Information_Complete	0x0C	Status, Connection_Handle, Version, Company_Identifier, Subversion

Description:

The HCI_Read_Remote_Version_Information_Complete 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 shall be for an ACL connection.

The Version event parameter defines the specification version of the BR/EDR or LE Controller. The Company_Identifier event parameter indicates the manufacturer of the remote Controller. The Subversion event parameter is controlled by the manufacturer and is vendor-specific. These parameters shall contain the same values as the CompanyID and SubVersion parameters in [\[Vol 2\] Part C, Section 4.3.3](#) and [\[Vol 6\] Part B, Section 2.4.2.13](#).

Event parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Remote_Version_Information command succeeded.
0x01 to 0xFF	HCI_Read_Remote_Version_Information command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Connection_Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

Version:

Size: 1 octet

Value	Parameter Description
0xFF	Version of the Current LMP or Link Layer supported by the remote Controller. See Assigned Numbers .

Host Controller Interface Functional Specification**Company_Identifier:****Size: 2 octets**

Value	Parameter Description
0xFFFF	Company identifier for the manufacturer of the remote Controller. See Assigned Numbers .

Subversion:**Size: 2 octets**

Value	Parameter Description
0xFFFF	Revision of the LMP or Link Layer implementation in the remote Controller. This value is vendor-specific.



7.7.13 QoS Setup Complete event

Event	Event Code	Event Parameters
HCI_QoS_Setup_Complete	0x0D	Status, Connection_Handle, Unused, Service_Type, Token_Rate, Peak_Bandwidth, Latency, Delay_Variation

Description:

The HCI_QoS_Setup_Complete event is used to indicate the completion of the process of the Link Manager setting up QoS with the remote BR/EDR Controller specified by the Connection_Handle event parameter. The Connection_Handle will be a Connection_Handle for an ACL connection. For more detail see [\[Vol 3\] Part A, Logical Link Control and Adaptation Protocol Specification](#).

The Unused parameter is reserved for future use.

This event is generated following an HCI_QoS_Setup command issued by the local Host.

Note: This event or the HCI_Flow_Specification_Complete event can be generated if the remote device performs an LMP transaction involving the flow parameter values.

Event parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_QoS_Setup command succeeded.
0x01 to 0xFF	HCI_QoS_Setup command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Connection_Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

*Unused:**Size: 1 octet*

Value	Parameter Description
0x00	This value shall be used by the Controller.
All other values	Reserved for future use.

*Service_Type:**Size: 1 octet*

Value	Parameter Description
0x00	No Traffic Available.
0x01	Best Effort Available.
0x02	Guaranteed Available.
All other values	Reserved for future use.

*Token_Rate:**Size: 4 octets*

Value	Parameter Description
0XXXXXXXX	Available Token Rate, in octets per second.

*Peak_Bandwidth:**Size: 4 octets*

Value	Parameter Description
0XXXXXXXX	Available Peak Bandwidth, in octets per second.

*Latency:**Size: 4 octets*

Value	Parameter Description
0XXXXXXXX	Available Latency, in microseconds.

*Delay_Variation:**Size: 4 octets*

Value	Parameter Description
0XXXXXXXX	Available Delay Variation, in microseconds.



7.7.14 Command Complete event

Event	Event Code	Event Parameters
HCI_Command_Complete	0x0E	Num_HCI_Command_Packets, Command_Opcode, Return_Parameters

Description:

The HCI_Command_Complete 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 the Host can send to the Controller. If the Controller requires the Host to stop sending commands, the Num_HCI_Command_Packets event parameter will be set to zero. To indicate to the Host that the Controller is ready to receive HCI command packets, the Controller generates an HCI_Command_Complete event with the Command_Opcode set to 0x0000 and the Num_HCI_Command_Packets event parameter set to 1 or more. Command_Opcode 0x0000 is a special value indicating that this event is not associated with a command sent by the Host. The Controller can send an HCI_Command_Complete event with Command Opcode 0x0000 at any time to change the number of outstanding HCI command packets that the Host can send before waiting. See each command for the parameters that are returned by this event.

Event parameters:

Num_HCI_Command_Packets:

Size: 1 octet

Value	Parameter Description
0xXX	The Number of HCI Command packets which are allowed to be sent to the Controller from the Host. Range: 0 to 255

Command_Opcode:

Size: 2 octets

Value	Parameter Description
0x0000	No associated command
0xFFFF	(non-zero) Opcode of the command which caused this event.

*Return_Parameters:**Size: Depends on command*

Value	Parameter Description
0xXX	This is the return parameter(s) for the command specified in the Command_Opcode event parameter. See each command's definition for the list of return parameters associated with that command.



7.7.15 Command Status event

Event	Event Code	Event Parameters
HCI_Command_Status	0x0F	Status, Num_HCI_Command_Packets, Command_Opcode

Description:

The HCI_Command_Status 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 avoids the need for the Host to wait 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 will contain the corresponding error code, and no complete event will follow since the command was not started. 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, the Num_HCI_Command_Packets event parameter will be set to zero. To indicate to the Host that the Controller is ready to receive HCI command packets, the Controller generates an HCI_Command_Status event with Status 0x00 and Command_Opcode 0x0000 and the Num_HCI_Command_Packets event parameter set to 1 or more. Command_Opcode 0x0000 is a special value indicating that this event is not associated with a command sent by the Host. The Controller can send an HCI_Command_Status event with Command Opcode 0x0000 at any time to change the number of outstanding HCI command packets that the Host can send before waiting.

Event parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	Command now pending.
0x01 to 0xFF	Command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Num_HCI_Command_Packets:

Size: 1 octet

Value	Parameter Description
0xXX	The Number of HCI Command packets which are allowed to be sent to the Controller from the Host. Range: 0 to 255

**Command_Opcode:****Size: 2 octets**

Value	Parameter Description
0x0000	No associated command
0xFFFF	(non-zero) Opcode of the command which caused this event and is now pending.



7.7.16 Hardware Error event

Event	Event Code	Event Parameters
HCI_Hardware_Error	0x10	Hardware_Code

Description:

The HCI_Hardware_Error event is used to notify the Host that a hardware failure has occurred in the Controller.

Event parameters:

Hardware_Code:

Size: 1 octet

Value	Parameter Description
0x00 to 0xFF	These Hardware_Codes will be implementation-specific, and can be assigned to indicate various hardware problems.



7.7.17 Flush Occurred event

Event	Event Code	Event Parameters
HCI_Flush_Occurred	0x11	Handle

Description:

The HCI_Flush_Occurred event is used to indicate that, for the specified Handle, the current user data to be transmitted has been removed. The Handle shall be a Connection_Handle for an ACL connection. This could result from the HCI_Flush command, or be due to the automatic flush. Multiple blocks of an L2CAP packet could have been pending in the Controller. If one Baseband packet part of an L2CAP PDU is flushed, then the rest of the HCI ACL Data packets for the L2CAP PDU shall also be flushed.

Event parameters:

Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xFFFF	Handle that was flushed. Range: 0x0000 to 0xFFFF



7.7.18 Role Change event

Event	Event Code	Event Parameters
HCI_Role_Change	0x12	Status, BD_ADDR, New_Role

Description:

The HCI_Role_Change event is used to indicate that the current role of the BR/EDR Controller related to the particular connection has changed. This event occurs (with Status set to zero) when both the remote and local BR/EDR Controllers have completed their role change for the BR/EDR Controller associated with the BD_ADDR event parameter, allowing both affected Hosts to be notified when the Role has been changed.

This event is also generated on the local Controller when the HCI_Switch_Role command finishes because no change was requested or an error occurred. If a Baseband role switch was attempted but failed, the remote Controller may generate the event.

Event parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	A role change has occurred.
0x01 to 0xFF	A role change failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0xFFFFFFFFXXXX	BD_ADDR of the Device for which a role change has completed.

New_Role:

Size: 1 octet

Value	Parameter Description
0x00	Currently the Central for specified BD_ADDR.
0x01	Currently the Peripheral for specified BD_ADDR.



7.7.19 Number Of Completed Packets event

Event	Event Code	Event Parameters
HCI_Number_Of_Completed_Packets	0x13	Num_Handles, Connection_Handle[i], Num_Completed_Packets[i]

Description:

The HCI_Number_Of_Completed_Packets event is used by the Controller to indicate to the Host how many HCI Data packets have been completed for each Connection_Handle since the previous HCI_Number_Of_Completed_Packets event was sent to the Host. This means that the corresponding buffer space has been freed in the Controller and is available for new packets to be sent. Based on this information and the return parameters of the HCI_Read_Buffer_Size and HCI_LE_Read_Buffer_Size commands, the Host can determine for which Connection_Handles the following HCI Data packets should be sent to the Controller. The HCI_Number_Of_Completed_Packets event shall not specify a given Connection_Handle before the event indicating the corresponding connection has been created or after an event indicating disconnection of the corresponding connection. While the Controller has HCI Data packets in its buffer, it shall keep sending the HCI_Number_Of_Completed_Packets event to the Host at least periodically, until it finally reports that all the pending packets have been completed. The rate with which this event is sent is manufacturer specific.

Note: HCI_Number_Of_Completed_Packets events will not report on synchronous Connection_Handles if synchronous Flow Control is disabled. (See [Section 7.3.36](#) and [Section 7.3.37](#).)

Event parameters:

Num_Handles:

Size: 1 octet

Value	Parameter Description
0xXX	The number of Connection_Handles and Num_HCI_Data_Packets parameters pairs contained in this event. Range: 0 to 255

Connection_Handle[i]:

Size: Num_Handles × 2 octets (12 bits meaningful)

Value	Parameter Description
0xFFFF	Connection_Handle. Range: 0x0000 to 0x0EFF



Num_Completed_Packets[i]:

Size: Num_Handles × 2 octets

Value	Parameter Description
0xFFFF	The number of HCI Data packets that have been completed for the associated Connection_Handle since the previous time the event was returned. Range: 0x0000 to 0xFFFF



7.7.20 Mode Change event

Event	Event Code	Event Parameters
HCI_Mode_Change	0x14	Status, Connection_Handle, Current_Mode, Interval

Description:

The HCI_Mode_Change event is used to indicate when the device associated with the Connection_Handle changes between Active mode, Hold mode, and Sniff mode. The Connection_Handle will be a Connection_Handle for an ACL connection. The Connection_Handle event parameter is used to indicate which connection the HCI_Mode_Change event is for. The Current_Mode event parameter is used to indicate which state the connection is currently in. The Interval parameter is used to specify a time amount specific to each state. Each Controller that is associated with the Connection_Handle which has changed modes shall send the HCI_Mode_Change event to its Host.

Event parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	A Mode Change has occurred.
0x01 to 0xFF	HCI_Hold_Mode, HCI_Sniff_Mode, or HCI_Exit_Sniff_Mode command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Connection_Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xFFFF	Connection_Handle. Range: 0x0000 to 0x0EFF

Current_Mode:

Size: 1 octet

Value	Parameter Description
0x00	Active mode.
0x01	Hold mode.
0x02	Sniff mode.
All other values	Reserved for future use.



Interval:

Size: 2 octets

Value	Parameter Description
N = 0xFFFF	Hold: Number of Baseband slots to wait in Hold mode. Hold Interval = N * 0.625 ms (1 Baseband slot) Range: 0x0002 to 0xFFFFE Time Range: 1.25 ms to 40.9 s Sniff: Number of Baseband slots between sniff anchor points. Time between sniff anchor points = N * 0.625 ms (1 Baseband slot) Range: 0x0002 to 0xFFFFE Time Range: 1.25 ms to 40.9 s



7.7.21 Return Link Keys event

Event	Event Code	Event Parameters
HCI_Return_Link_Keys	0x15	Num_Keys, BD_ADDR[i], Link_Key[i]

Description:

The HCI_Return_Link_Keys event is used by the BR/EDR Controller to send the Host the BD_ADDRs associated with one or more stored Link Keys. Zero or more instances of this event will occur after the HCI_Read_Stored_Link_Key command. When there are no link keys stored, no HCI_Return_Link_Keys events shall be returned. When there are link keys stored, the number of link keys returned in each HCI_Return_Link_Keys event is implementation specific. This event shall never return the value of the link keys. The link keys value parameter shall always contain the value of zero.

Event parameters:

Num_Keys:

Size: 1 octet

Value	Parameter Description
0xXX	Number of Link Keys contained in this event. Range: 0x01 to 0x0B

BD_ADDR[i]:

Size: Num_Keys × 6 octets

Value	Parameter Description
0XXXXXXXXXXXXX	BD_ADDR for the associated Link Key.

Link_Key[i]:

Size: Num_Keys × 16 octets

Value	Parameter Description
0x000000000000 00000000000000 000000	Shall be zero.



7.7.22 PIN Code Request event

Event	Event Code	Event Parameters
HCI_PIN_Code_Request	0x16	BD_ADDR

Description:

The HCI_PIN_Code_Request event is used to indicate that a PIN code is required to create a new link key. The Host shall respond using either the HCI_PIN_Code_Request_Reply or the HCI_PIN_Code_Request_Negative_Reply command, depending on whether the Host can provide the Controller with a PIN code or not.

Note: If the HCI_PIN_Code_Request event is masked away, then the BR/EDR Controller will assume that the Host has no PIN Code.

When the BR/EDR Controller generates an HCI_PIN_Code_Request event in order for the local Link Manager to respond to the request from the remote Link Manager (as a result of an HCI_Create_Connection or HCI_Authentication_Requested command from the remote Host), the local Host shall respond with either an HCI_PIN_Code_Request_Reply or HCI_PIN_Code_Request_Negative_Reply command before the remote Link Manager detects LMP response timeout. (See [\[Vol 2\] Part C, Link Manager Protocol Specification](#).)

Event parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0XXXXXXXXXXXXX	BD_ADDR of the Device which a new link key is being created for.



7.7.23 Link Key Request event

Event	Event Code	Event Parameters
HCI_Link_Key_Request	0x17	BD_ADDR

Description:

The HCI_Link_Key_Request event is used to indicate that a Link Key is required for the connection with the device specified in BD_ADDR. If the Host has the requested stored Link Key, then the Host shall pass the requested Key to the Controller using the HCI_Link_Key_Request_Reply command. If the Host does not have the requested stored Link Key, or the stored Link Key does not meet the security requirements for the requested service, then the Host shall use the HCI_Link_Key_Request_Negative_Reply command to indicate to the Controller that the Host does not have the requested key.

Note: If the HCI_Link_Key_Request event is masked away, then the BR/EDR Controller will assume that the Host has no additional link keys.

If the Host uses the HCI_Link_Key_Request_Negative_Reply command when the requested service requires an authenticated Link Key and the current Link Key is unauthenticated, the Host should set the Authentication_Requirements parameter one of the MITM Protection Required options.

When the Controller generates an HCI_Link_Key_Request event in order for the local Link Manager to respond to the request from the remote Link Manager (as a result of an HCI_Create_Connection or HCI_Authentication_Requested command from the remote Host), the local Host shall respond with either an HCI_Link_Key_Request_Reply or HCI_Link_Key_Request_Negative_Reply command before the remote Link Manager detects LMP response timeout. (See [\[Vol 2\] Part C, Link Manager Protocol Specification](#).)

Event parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0xFFFFFFFFXXXX	BD_ADDR of the Device which a stored link key is being requested.



7.7.24 Link Key Notification event

Event	Event Code	Event Parameters
HCI_Link_Key_Notification	0x18	BD_ADDR, Link_Key, Key_Type

Description:

The HCI_Link_Key_Notification event is used to indicate to the Host that a new Link Key has been created for the connection with the device specified in BD_ADDR. The Host can save this new Link Key in its own storage for future use. Also, the Host can decide to store the Link Key in the BR/EDR Controller's Link Key Storage by using the HCI_Write_Stored_Link_Key command. The Key_Type event parameter informs the Host about which key type (combination key, debug combination key, unauthenticated combination key, authenticated combination key or changed combination key) was used during pairing. If the key type is not supported or is reserved for future use, the Host may discard the key or disconnect the link.

The combination key Key_Type is used when standard pairing was used. The debug combination key Key_Type is used when Secure Simple Pairing was used and the debug public key is sent or received. The unauthenticated combination key Key_Type is used when the Just Works Secure Simple Pairing association model was used. The authenticated combination key Key_Type is used when Secure Simple Pairing was used and the Just Works association mode was not used. The changed combination key Key_Type is used when the link key has been changed using the Change Connection Link Key procedure and Secure Simple Pairing Mode is set to enabled.

Note: It is the responsibility of the Host to remember the Key_Type (combination, debug combination, unauthenticated combination, or authenticated combination) prior to changing the link key.

When the unauthenticated or authenticated combination key Key_Type is used, the Controller shall use values 0x04 and 0x05 to indicate keys created with the P-192 elliptic curve and values 0x07 and 0x08 to indicate keys created with the P-256 elliptic curve. The values 0x07 and 0x08 shall only be used when the Host has indicated support for Secure Connections in the Secure_Connections_Host_Support parameter.

Event parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0xFFFFFFFFXXXX	BD_ADDR of the Device for which the new link key has been generated.

**Link_Key:****Size: 16 octets**

Value	Parameter Description
0xFFFFFFFFXXXXXXX XXXXXXXXXXXX XXXXXXXXXXXX	Link Key for the associated BD_ADDR.

Key_Type:**Size: 1 octet**

Value	Parameter Description
0x00	Combination Key
0x03	Debug Combination Key
0x04	Unauthenticated Combination Key generated from P-192
0x05	Authenticated Combination Key generated from P-192
0x06	Changed Combination Key
0x07	Unauthenticated Combination Key generated from P-256
0x08	Authenticated Combination Key generated from P-256
All other values	Reserved for future use



7.7.25 Loopback Command event

Event	Event Code	Event Parameters
HCI_Loopback_Command	0x19	HCI_Command_Packet

Description:

When in Local Loopback mode, the BR/EDR Controller loops back commands and data to the Host. The HCI_Loopback_Command event is used to loop back all commands that the Host sends to the Controller with some exceptions. See [Section 7.6.1](#) for a description of which commands that are not looped back. The HCI_Command_Packet event parameter contains the entire HCI Command Packet including the header.

Note: The event packet is limited to a maximum of 255 octets in the payload; since an HCI Command packet has 3 octets of header data, only the first 252 octets of the command parameters will be returned.

Event parameters:

HCI_Command_Packet:

Size: Depends on command

Value	Parameter Description
0xXXXXXX	HCI Command packet, including header.



7.7.26 Data Buffer Overflow event

Event	Event Code	Event Parameters
HCI_Data_Buffer_Overflow	0x1A	Link_Type

Description:

This event is used to indicate that the Controller's data buffers have been overflowed. This can occur if the Host has sent more packets than allowed. The Link_Type parameter is used to indicate the type of data whose buffers overflowed.

Event parameters:

Link_Type:

Size: 1 octet

Value	Parameter Description
0x00	Synchronous Data packet buffers
0x01	ACL Data packet buffers
0x02	ISO Data packet buffers
All other values	Reserved for future use.



7.7.27 Max Slots Change event

Event	Event Code	Event Parameters
HCI_Max_Slots_Change	0x1B	Connection_Handle, LMP_Max_Slots

Description:

This event is used to notify the Host about the LMP_Max_Slots parameter when the value of this parameter changes. It shall be sent each time the maximum allowed length, in number of slots, for Baseband packets transmitted by the local device, changes. The Connection_Handle will be a Connection_Handle for an ACL connection.

Event parameters:

Connection_Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xXXXX	Connection_Handle. Range: 0x0000 to 0x0EFF

LMP_Max_Slots: *Size: 1 octet*

Value	Parameter Description
0x01, 0x03, 0x05	Maximum number of slots allowed to use for Baseband packets, see [Vol 2] Part C, Section 4.1.10 and Section 5.2 .



7.7.28 Read Clock Offset Complete event

Event	Event Code	Event Parameters
HCI_Read_Clock_Offset_Complete	0x1C	Status, Connection_Handle, Clock_Offset

Description:

The HCI_Read_Clock_Offset_Complete event is used to indicate the completion of the process of the Link Manager obtaining the Clock Offset information of the BR/EDR Controller specified by the Connection_Handle event parameter. The Connection_Handle will be a Connection_Handle for an ACL connection.

Event parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Read_Clock_Offset command succeeded.
0x01 to 0xFF	HCI_Read_Clock_Offset command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Connection_Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

Clock_Offset:

Size: 2 octets

Bit Number	Parameter Description
0-14	Bits 16-2 of CLKNPeripheral - CLK
15	Reserved for future use.



7.7.29 Connection Packet Type Changed event

Event	Event Code	Event Parameters
HCI_Connection_Packet_Type_Changed	0x1D	Status, Connection_Handle, Packet_Type

Description:

The HCI_Connection_Packet_Type_Changed event is used to indicate that the process has completed of the Link Manager changing which packet types can be used for the connection. This allows current connections to be dynamically modified to support different types of user data. The Packet_Type event parameter specifies which packet types the Link Manager can use for the connection identified by the Connection_Handle event parameter for sending L2CAP data or voice. The Packet_Type event parameter does not decide which packet types the LM is allowed to use for sending LMP PDUs.

Event parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Change_Connection_Packet_Type command succeeded.
0x01 to 0xFF	HCI_Change_Connection_Packet_Type command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Connection_Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xFFFF	Connection_Handle. Range: 0x0000 to 0x0EFF

Packet_Type:

Size: 2 octets

For ACL_Link_Type

Bit Number	Parameter Description
1	2-DH1 shall not be used.
2	3-DH1 shall not be used.
3	Ignored; DM1 may be used whether or not this bit is set.
4	DH1 may be used.
8	2-DH3 shall not be used.
9	3-DH3 shall not be used.
10	DM3 may be used.



Bit Number	Parameter Description
11	DH3 may be used.
12	2-DH5 shall not be used.
13	3-DH5 shall not be used.
14	DM5 may be used.
15	DH5 may be used.
All other bits	Reserved for future use.

For SCO_Link_Type

Bit Number	Parameter Description
5	HV1 may be used.
6	HV2 may be used.
7	HV3 may be used.
All other bits	Reserved for future use.



7.7.30 QoS Violation event

Event	Event Code	Event Parameters
HCI_QoS_Violation	0x1E	Handle

Description:

The HCI_QoS_Violation event is used to indicate the Controller is unable to provide the current QoS requirement for the Connection identified by the Handle. This event indicates that the Controller is unable to provide one or more of the agreed QoS parameters.

The Host chooses what action should be done; for example, it can reissue the HCI_QoS_Setup command to renegotiate the QoS setting for Connection_Handle.

Event parameters:

Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xXXXX	Handle for the link that the Controller cannot provide the current QoS requested. The Handle is a Connection_Handle for a BR/EDR Controller. Range: 0x0000 to 0x0EFF



7.7.31 Page Scan Repetition Mode Change event

Event	Event Code	Event Parameters
HCI_Page_Scan_Repetition_Mode_Change	0x20	BD_ADDR, Page_Scan_Repetition_Mode

Description:

The HCI_Page_Scan_Repetition_Mode_Change event indicates that the remote BR/EDR Controller with the specified BD_ADDR has successfully changed the Page Scan Repetition Mode (see [\[Vol 2\] Part B, Section 8.3.1](#)).

Event parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0XXXXXXXXX XXXX	BD_ADDR of the remote device.

Page_Scan_Repetition_Mode:

Size: 1 octet

Value	Parameter Description
0x00	R0
0x01	R1
0x02	R2
All other values	Reserved for future use



7.7.32 Flow Specification Complete event

Event	Event Code	Event Parameters
HCI_Flow_Specification_Complete	0x21	Status, Connection_Handle, Unused, Flow_Direction, Service_Type, Token_Rate, Token_Bucket_Size, Peak_Bandwidth, Access_Latency

Description:

The HCI_Flow_Specification_Complete event is used to inform the Host about the Quality of Service for the ACL connection the Controller is able to support. The Connection_Handle will be a Connection_Handle for an ACL connection. The flow parameters refer to the outgoing or incoming traffic of the ACL link, as indicated by the Flow_Direction field. The flow parameters are defined in the L2CAP specification [\[Vol 3\] Part A, Section 5.3](#). When the Status parameter indicates a successful completion, the flow parameters specify the agreed values by the Controller. When the Status parameter indicates a failed completion with the error code *QoS Unacceptable Parameter* (0x2C), the flow parameters specify the acceptable values of the Controller. This enables the Host to continue the 'QoS negotiation' with a new HCI_Flow_Specification command with flow parameter values that are acceptable for the Controller. When the Status parameter indicates a failed completion with the error code *QoS Rejected* (0x2D), this indicates a request of the Controller to discontinue the 'QoS negotiation'. When the Status parameter indicates a failed completion, the flow parameter values of the most recently successful completion shall be assumed (or the default values when there was no success completion).

The Unused parameter is reserved for future use.

This event is generated following an HCI_Flow_Specification command issued by the local Host.

Note: This event or the HCI_QoS_Setup_Complete event can be generated if the remote device performs an LMP transaction involving the flow parameter values.

**Event parameters:***Status:**Size: 1 octet*

Value	Parameter Description
0x00	HCI_Flow_Specification command succeeded
0x01 to 0xFF	HCI_Flow_Specification command failed. See [Vol 1] Part F, Controller Error Codes for list of error codes

*Connection_Handle:**Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

*Unused:**Size: 1 octet*

Value	Parameter Description
0x00	This value shall be used by the Controller.
All other values	Reserved for future use.

*Flow_Direction:**Size: 1 octet*

Value	Parameter Description
0x00	Outgoing Flow i.e., traffic sent over the ACL connection.
0x01	Incoming Flow i.e., traffic received over the ACL connection.
All other values	Reserved for future use.

*Service_Type:**Size: 1 octet*

Value	Parameter Description
0x00	No Traffic
0x01	Best Effort
0x02	Guaranteed
All other values	Reserved for future use

*Token_Rate:**Size: 4 octets*

Value	Parameter Description
0xFFFFFFFF	Token Rate in octets per second

***Token_Bucket_Size:******Size: 4 octets***

Value	Parameter Description
0xFFFFFFFF	Token Bucket Size in octets

Peak_Bandwidth:***Size: 4 octets***

Value	Parameter Description
0xFFFFFFFF	Peak Bandwidth in octets per second

Access_Latency:***Size: 4 octets***

Value	Parameter Description
0xFFFFFFFF	Access Latency in microseconds



7.7.33 Inquiry Result with RSSI event

Event	Event Code	Event Parameters
HCI_Inquiry_Result_with_RSSI	0x22	Num_Responses, BD_ADDR[i], Page_Scan_Repetition_Mode[i], Reserved[i], Class_Of_Device[i], Clock_Offset[i], RSSI[i]

Description:

The HCI_Inquiry_Result_with_RSSI event indicates that a BR/EDR Controller or multiple BR/EDR Controllers have responded so far during the current Inquiry process. This event will be sent from the BR/EDR Controller to the Host as soon as an Inquiry Response from a remote device is received if the remote device supports only mandatory paging scheme. This BR/EDR Controller may queue these Inquiry Responses and send multiple BR/EDR Controllers information in one HCI_Inquiry_Result event. The event can be used to return one or more Inquiry responses in one event. The RSSI parameter is measured during the FHS packet returned by each responding Peripheral.

This event shall only be generated if the Inquiry Mode parameter of the last HCI_Write_Inquiry_Mode command was set to 0x01 (Inquiry Result format with RSSI), or was set to 0x02 (Inquiry Result with RSSI format or Extended Inquiry Result format) and the inquiry response packet had the EIR field set to 0.

Event parameters:

Num_Responses:

Size: 1 octet

Value	Parameter Description
0xXX	Number of responses from the Inquiry.

BD_ADDR[i]:

Size: Num_Responses × 6 octets

Value	Parameter Description
0XXXXXXXXXX XX	BD_ADDR for a device which responded.

*Page_Scan_Repetition_Mode[i]:**Size: Num_Responses × 1 octet*

Value	Parameter Description
0x00	R0
0x01	R1
0x02	R2
All other values	Reserved for future use

*Reserved[i]:**Size: Num_Responses × 1 octet*

Value	Parameter Description
0xXX	Reserved for future use.

*Class_Of_Device[i]:**Size: Num_Responses × 3 octets*

Value	Parameter Description
0XXXXXXX	Class of Device for the device

*Clock_Offset[i]:**Size: Num_Responses × 2 octets*

Bit Number	Parameter Description
0-14	Bits 16-2 of CLKNPeripheral - CLK
15	Reserved for future use

*RSSI[i]:**Size: Num_Responses × 1 octet*

Value	Parameter Description
0xXX	Range: -127 to +20 Units: dBm



7.7.34 Read Remote Extended Features Complete event

Event	Event Code	Event Parameters
HCI_Read_Remote_Extended_Features_Complete	0x23	Status, Connection_Handle, Page_Number, Max_Page_Number, Extended_LMP_Features

Description:

The HCI_Read_Remote_Extended_Features_Complete event is used to indicate the completion of the process of the Link Manager obtaining the remote extended LMP features of the remote device specified by the Connection_Handle event parameter. The Connection_Handle will be a Connection_Handle for an ACL connection. The event parameters include a page of the remote devices extended LMP features. For details see [\[Vol 2\] Part C, Link Manager Protocol Specification](#).

Note: If a feature page is requested more than once while a connection exists between the two devices, the second and subsequent requests may report a cached copy of that page rather than fetching it again.

Event parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	Request for remote extended features succeeded
0x01 to 0xFF	Request for remote extended features failed. See [Vol 1] Part F, Controller Error Codes for error codes.

Connection_Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

Page_Number:

Size: 1 octet

Value	Parameter Description
0x00	The normal LMP features as returned by HCI_Read_Remote_Supported_Features command
0x01 to 0xFF	The page number of the features returned

Host Controller Interface Functional Specification**Max_Page_Number:****Size: 1 octet**

Value	Parameter Description
0x00 to 0xFF	The highest features page number which contains non-zero bits for the remote device

Extended_LMP_Features:**Size: 8 octets**

Value	Parameter Description
0XXXXXXXXXXXXXXXXX	Bit map of requested page of LMP features. See LMP [Vol 2] Part C, Section 3.3 for details.



7.7.35 Synchronous Connection Complete event

Event	Event Code	Event Parameters
HCI_Synchronous_ - Connection_Complete	0x2C	Status, Connection_Handle, BD_ADDR, Link_Type, Transmission_Interval, Retransmission_Window, RX_Packet_Length, TX_Packet_Length, Air_Mode

Description:

The HCI_Synchronous_Connection_Complete event is sent to indicate completion of any of the following commands:

- HCI_Setup_Synchronous_Connection
- HCI_Accept_Synchronous_Connection_Request
- HCI_Reject_Synchronous_Connection_Request
- HCI_Enhanced_Setup_Synchronous_Connection
- HCI_Enhanced_Accept_Synchronous_Connection_Request

This event returns the completion status for the command.

When the HCI_Synchronous_Connection_Complete event was triggered by the HCI_Enhanced_Setup_Synchronous_Connection or HCI_Enhanced_Accept_Synchronous_Connection_Request commands, the Controller shall set the Air_Mode parameter to the Transmit Air Coding Format parameter of the original command.

Event parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	Connection successfully completed.
0x01 to 0xFF	Connection failed to complete. See [Vol 1] Part F, Controller Error Codes for error codes and descriptions.

**Connection_Handle:****Size: 2 octets (12 bits meaningful)**

Value	Parameter Description
0xXXXX	Connection_Handle Range: 0x0000 to 0x0EFF

BD_ADDR:**Size: 6 octets**

Value	Parameter Description
0XXXXXXXXXXXXX	BD_ADDR of the other connected device forming the connection.

Link_Type:**Size: 1 octet**

Value	Parameter Description
0x00	SCO connection
0x01	Reserved for future use
0x02	eSCO connection
All other values	Reserved for future use

Transmission_Interval:**Size: 1 octet**

Value	Parameter Description
0xXX	Time between two consecutive eSCO instants measured in slots. Shall be zero for SCO links.

Retransmission_Window:**Size: 1 octet**

Value	Parameter Description
0xXX	The size of the retransmission window measured in slots. Shall be zero for SCO links.

RX_Packet_Length:**Size: 2 octets**

Value	Parameter Description
0XXXXX	Length in bytes of the eSCO payload in the receive direction. Shall be zero for SCO links.

TX_Packet_Length:**Size: 2 octets**

Value	Parameter Description
0XXXXX	Length in bytes of the eSCO payload in the transmit direction. Shall be zero for SCO links.

***Air_Mode:******Size: 1 octet***

Value	Parameter Description
0x00	μ -law log
0x01	A-law log
0x02	CVSD
0x03	Transparent Data
All other values	Reserved for future use



7.7.36 Synchronous Connection Changed event

Event	Event Code	Event Parameters
HCI_Synchronous_Connection_Changed	0x2D	Status, Connection_Handle, Transmission_Interval, Retransmission_Window, RX_Packet_Length, TX_Packet_Length

Description:

The HCI_Synchronous_Connection_Changed event indicates to the Host that an existing synchronous connection has been reconfigured. This event also indicates to the initiating Host (if the change was Host initiated) if the issued command failed or was successful.

Event parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	Connection successfully reconfigured.
0x01 to 0xFF	Reconfiguration failed to complete. See [Vol 1] Part F, Controller Error Codes for error codes and descriptions.

Connection_Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

Transmission_Interval:

Size: 1 octet

Value	Parameter Description
0xFF	Time between two consecutive SCO/eSCO instants measured in slots.

Retransmission_Window:

Size: 1 octet

Value	Parameter Description
0xFF	The size of the retransmission window measured in slots. Shall be zero for SCO links.

Host Controller Interface Functional Specification***RX_Packet_Length:******Size: 2 octets***

Value	Parameter Description
0xFFFF	Length in bytes of the SCO/eSCO payload in the receive direction.

TX_Packet_Length:***Size: 2 octets***

Value	Parameter Description
0xFFFF	Length in bytes of the SCO/eSCO payload in the transmit direction.



7.7.37 Sniff Subrating event

Event	Event Code	Event Parameters
HCI_Sniff_Subrating	0x2E	Status, Connection_Handle, Max_TX_Latency, Max_RX_Latency, Min_Remote_Timeout, Min_Local_Timeout

Description:

The HCI_Sniff_Subrating event indicates that the device associated with the Connection_Handle has either enabled sniff subrating or the sniff subrating parameters have been renegotiated by the link manager. The Connection_Handle will be a Connection_Handle for an ACL connection. The Connection_Handle event parameter indicates which connection the HCI_Sniff_Subrating event is for.

Each BR/EDR Controller that is associated with the Connection_Handle that has changed its subrating parameters will send the Sniff Subrating event to its Host.

Event parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Sniff_Subrating command succeeded
0x01 to 0xFF	HCI_Sniff_Subrating command failed to complete. See [Vol 1] Part F, Controller Error Codes for error codes and descriptions.

Connection_Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xFFFF	Connection_Handle. Range: 0x0000 to 0x0EFF

**Max_TX_Latency:****Size: 2 octets**

Value	Parameter Description
N = 0xFFFF	Maximum latency for data being transmitted from the local device to the remote device. Latency = $N * 0.625$ ms (1 Baseband slot) Range: 0x0000 to 0xFFFFE Time Range: 0 s to 40.9 s

Max_RX_Latency:**Size: 2 octets**

Value	Parameter Description
N = 0xFFFF	Maximum latency for data being received by the local device from the remote device. Latency = $N * 0.625$ ms (1 Baseband slot) Range: 0x0000 to 0xFFFFE Time Range: 0 s to 40.9 s

Min_Remote_Timeout:**Size: 2 octets**

Value	Parameter Description
N = 0xFFFF	The base sniff subrate timeout in Baseband slots that the remote device shall use. Timeout = $N * 0.625$ ms (1 Baseband slot) Range: 0x0000 to 0xFFFFE Time Range: 0 s to 40.9 s

Min_Local_Timeout:**Size: 2 octets**

Value	Parameter Description
N = 0xFFFF	The base sniff subrate timeout in Baseband slots that the local device will use. Timeout = $N * 0.625$ ms (1 Baseband slot) Range: 0x0000 to 0xFFFFE Time Range: 0 s to 40.9 s



7.7.38 Extended Inquiry Result event

Event	Event Code	Event Parameters
HCI_Extended_Inquiry_ - Result	0x2F	Num_Responses, BD_ADDR, Page_Scan_Repetition_Mode, Reserved, Class_Of_Device, Clock_Offset, RSSI, Extended_Inquiry_Response

Description:

The HCI_Extended_Inquiry_Result event indicates that a BR/EDR Controller has responded during the current inquiry process with extended inquiry response data. This event will be sent from the BR/EDR Controller to the Host upon reception of an Extended Inquiry Response from a remote device. One single Extended Inquiry Response is returned per event. This event contains RSSI and inquiry response data for the BR/EDR Controller that responded to the latest inquiry. The RSSI parameter is measured during the FHS packet returned by each responding Peripheral. The Num_Responses parameter shall be set to one.

This event is only generated if the Inquiry_Mode parameter of the last HCI_Write_Inquiry_Mode command was set to 0x02 (Inquiry Result with RSSI format or Extended Inquiry Result format).

Note: This ensures that a Host that does not support Extended Inquiry Results will never receive the HCI_Extended_Inquiry_Result event.

If an inquiry response packet with the EIR field set to zero is received, the HCI_Inquiry_Result_with_RSSI event format shall be used. If the EIR bit is set to one the HCI_Extended_Inquiry_Result event format shall be used. If the EIR bit is set to one but the Controller failed to receive the extended inquiry response packet, the Extended_Inquiry_Response parameter is set to zeros. If an extended inquiry response packet from the same device is correctly received in a later response, another event shall be generated.

Note: The only difference between the HCI_Extended_Inquiry_Result event and the HCI_Inquiry_Result_with_RSSI event is the additional Extended_Inquiry_Response parameter.

Note: The Extended_Inquiry_Response parameter is not interpreted by the Controller. The tagged data set by the other Host should be passed unaltered if it has been correctly received.

**Event parameters:***Num_Responses:**Size: 1 octet*

Value	Parameter Description
0x01	Number of responses from the inquiry. The HCI_Extended_Inquiry_Result event always contains a single response.

*BD_ADDR:**Size: 6 octets*

Value	Parameter Description
0XXXXXXXXX XXXX	BD_ADDR for the device that responded.

*Page_Scan_Repetition_Mode:**Size: 1 octet*

Value	Parameter Description
0x00	R0
0x01	R1
0x02	R2
All other values	Reserved for future use

*Reserved:**Size: 1 octet*

Value	Parameter Description
0xXX	Reserved for future use.

*Class_Of_Device:**Size: 3 octets*

Value	Parameter Description
0XXXXXXX	Class of Device for the device that responded.

*Clock_Offset:**Size: 2 octets*

Bit Number	Parameter Description
0-14	Bits 16-2 of CLKNPeripheral - CLK
15	Reserved for future use.

*RSSI:**Size: 1 octet*

Value	Parameter Description
0xXX	Range: -127 to +20 Units: dBm



Extended_Inquiry_Response:

Size: 240 octets

Value	Parameter Description
	Extended Inquiry Response data as defined in [Vol 3] Part C, Section 8



7.7.39 Encryption Key Refresh Complete event

Event	Event Code	Event Parameters
HCI_Encryption_Key_Refresh_Complete	0x30	Status, Connection_Handle

Description:

The HCI_Encryption_Key_Refresh_Complete event is used to indicate to the Host that the encryption key was refreshed on the given Connection_Handle any time encryption is paused and then resumed. The Controller shall send this event when the encryption key has been refreshed due to encryption being started or resumed.

If the HCI_Encryption_Key_Refresh_Complete event was generated due to an encryption pause and resume operation embedded within a change connection link key procedure, the HCI_Encryption_Key_Refresh_Complete event shall be sent prior to the HCI_Change_Connection_Link_Key_Complete event.

If the HCI_Encryption_Key_Refresh_Complete event was generated due to an encryption pause and resume operation embedded within a role switch procedure, the HCI_Encryption_Key_Refresh_Complete event shall be sent prior to the HCI_Role_Change event.

Event parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	Encryption key refresh completed successfully
0x01 to 0xFF	Encryption key refresh failed. See [Vol 1] Part F, Controller Error Codes for list of error codes

Connection_Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF



7.7.40 IO Capability Request event

Event	Event Code	Event Parameters
HCI_IO_Capability_Request	0x31	BD_ADDR

Description:

The HCI_IO_Capability_Request event is used to indicate that the IO capabilities of the Host are required for a Secure Simple Pairing process. The Host shall respond with an HCI_IO_Capability_Request_Reply command or HCI_IO_Capability_Request_Negative_Reply command. This event shall only be generated if Secure Simple Pairing has been enabled with the HCI_Write_Simple_Pairing_Mode command.

Event parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0xFFFFFFFFXXXX	BD_ADDR of remote device involved in the Secure Simple Pairing process



7.7.41 IO Capability Response event

Event	Event Code	Event Parameters
HCI_IO_Capability_Response	0x32	BD_ADDR, IO_Capability, OOB_Data_Present, Authentication_Requirements

Description:

The HCI_IO_Capability_Response event is used to indicate to the Host that IO capabilities from a remote device specified by BD_ADDR have been received during a Secure Simple Pairing process. This event will only be generated if Secure Simple Pairing has been enabled with the HCI_Write_Simple_Pairing_Mode command.

Event parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0xFFFFFFFFXXXX	BD_ADDR identifying the device to which the IO capabilities apply.

IO_Capability:

Size: 1 octet

Value	Parameter Description
0x00	DisplayOnly
0x01	DisplayYesNo
0x02	KeyboardOnly
0x03	NoInputNoOutput
All other values	Reserved for future use

OOB_Data_Present:

Size: 1 octet

Value	Parameter Description
0x00	OOB authentication data not present
0x01	OOB authentication data from remote device present
All other values	Reserved for future use

**Authentication_Requirements:****Size: 1 octet**

Value	Parameter Description
0x00	MITM Protection Not Required – No Bonding. Numeric comparison with automatic accept allowed.
0x01	MITM Protection Required – No Bonding. Use IO Capabilities to determine authentication procedure
0x02	MITM Protection Not Required – Dedicated Bonding. Numeric comparison with automatic accept allowed.
0x03	MITM Protection Required – Dedicated Bonding. Use IO Capabilities to determine authentication procedure
0x04	MITM Protection Not Required – General Bonding. Numeric Comparison with automatic accept allowed.
0x05	MITM Protection Required – General Bonding. Use IO capabilities to determine authentication procedure.
All other values	Reserved for future use



7.7.42 User Confirmation Request event

Event	Event Code	Event Parameters
HCI_User_Confirmation_Request	0x33	BD_ADDR, Numeric_Value

Description:

The HCI_User_Confirmation_Request event is used to indicate that user confirmation of a numeric value is required. The Host shall reply with either the HCI_User_Confirmation_Request_Reply or the HCI_User_Confirmation_Request_Negative_Reply command. If the Host has output capability (DisplayYesNo or KeyboardOnly), it shall display the Numeric_Value until the HCI_Simple_Pairing_Complete event is received. It shall reply based on the yes/no response from the user. If the Host has no input and no output it shall reply with the HCI_User_Confirmation_Request_Reply command. When the Controller generates an HCI_User_Confirmation_Request event, in order for the local Link Manager to respond to the request from the remote Link Manager, the local Host shall respond with either an HCI_User_Confirmation_Request_Reply or an HCI_User_Confirmation_Request_Negative_Reply command before the remote Link Manager detects LMP response timeout. (See [Vol 2] Part C, Link Manager Protocol Specification.)

Event parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0xFFFFFFFFXXXX	BD_ADDR of device involved in the Secure Simple Pairing process.

Numeric_Value:

Size: 4 octets

Value	Parameter Description
0x00000000 to 0x000F423F	Numeric value to be displayed. Valid values are decimal 000000 to 999999.



7.7.43 User Passkey Request event

Event	Event Code	Event Parameters
HCI_User_Passkey_Request	0x34	BD_ADDR

Description:

The HCI_User_Passkey_Request event is used to indicate that a passkey is required as part of a Secure Simple Pairing process. The Host shall respond with either an HCI_User_Passkey_Request_Reply or HCI_User_Passkey_Request_Negative_Reply command. This event will only be generated if Secure Simple Pairing has been enabled with the HCI_Write_Simple_Pairing_Mode command. When the Controller generates an HCI_User_Passkey_Request event, in order for the local Link Manager to respond to the request from the remote Link Manager, the local Host shall respond with either an HCI_User_Passkey_Request_Reply or HCI_User_Passkey_Request_Negative_Reply command before the remote Link Manager detects LMP response timeout. (See [\[Vol 2\] Part C, Link Manager Protocol Specification](#).)

Event parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0XXXXXXXXXXXXX	BD_ADDR of the device involved in the Secure Simple Pairing process.



7.7.44 Remote OOB Data Request event

Event	Event Code	Event Parameters
HCI_Remote_OOB_Data_Request	0x35	BD_ADDR

Description:

The HCI_Remote_OOB_Data_Request event is used to indicate that the Secure Simple Pairing Hash C and Randomizer R are required for the Secure Simple Pairing process involving the device identified by BD_ADDR. The C and R values were transferred to the Host from the remote device via an OOB mechanism. This event is sent by the Controller because the Host previously set the OOB Data Present parameter to "OOB authentication data from remote device present" in an HCI_IO_Capability_Request_Reply command. If both the Host and Controller support Secure Connections the Host shall respond with the values using the HCI_Remote_OOB_Extended_Data_Request_Reply command. Otherwise, the Host shall respond with the values using the HCI_Remote_OOB_Data_Request_Reply command. In either case, if the Host does not have the C and R values for the device, it shall respond with the HCI_Remote_OOB_Data_Request_Negative_Reply command.

Event parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0xFFFFFFFFXXXX	BD_ADDR of the device from which the C and R values were received.



7.7.45 Simple Pairing Complete event

Event	Event Code	Event Parameters
HCI_Simple_Pairing_Complete	0x36	Status, BD_ADDR

Description:

The HCI_Simple_Pairing_Complete event is used to indicate that the Secure Simple Pairing process has completed. A Host that is displaying a numeric value can use this event to change its UI.

When the LMP Secure Simple Pairing sequences fail for any reason, the HCI_Simple_Pairing_Complete event shall be sent to the Host. When HCI_Simple_Pairing_Complete event is sent in response to the IO capability exchange failing, the Status parameter shall be set to the error code received from the remote device. Otherwise, the Status shall be set to the error code *Authentication Failure* (0x05).

Event parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	Secure Simple Pairing succeeded
0x01 to 0xFF	Secure Simple Pairing failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes.

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0XXXXXXXXXXXXX	BD_ADDR of the device involved in the Secure Simple Pairing process.



7.7.46 Link Supervision Timeout Changed event

Event	Event Code	Event Parameters
HCI_Link_Supervision_Timeout_Changed	0x38	Connection_Handle, Link_Supervision_Timeout

Description:

The HCI_Link_Supervision_Timeout_Changed event is used to notify the Peripheral's Host when the Link_Supervision_Timeout parameter is changed in the Peripheral's Controller. This event shall only be sent to the Host by the Peripheral's Controller upon receiving an LMP_SUPERVISION_TIMEOUT PDU from the Central.

Note: The Connection_Handle used for this command shall be the ACL connection of the appropriate device.

Event parameters:

Connection_Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xXXXX	Connection_Handle Range: 0x0000 to 0x0EFF

Link_Supervision_Timeout:

Size: 2 octets

Value	Parameter Description
N = 0xXXXX	Measured in number of Baseband slots $\text{Link_Supervision_Timeout} = N \times 0.625 \text{ ms}$ (1 Baseband slot) Range: 0x0001 to 0xFFFF Time Range: 0.625 ms to 40.9 s (0 means infinite timeout)



7.7.47 Enhanced Flush Complete event

Event	Event Code	Event Parameters
HCI_Enhanced_Flush_Complete	0x39	Handle

Description:

The HCI_Enhanced_Flush_Complete event is used to indicate that an Enhanced Flush is complete for the specified Handle.

Event parameters:

Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xFFFF	Handle of the connection for which the Enhanced Flush was done. The Handle is a Connection_Handle for a BR/EDR Controller. Range: 0x0000 to 0x0EFF



7.7.48 User Passkey Notification event

Event	Event Code	Event Parameters
HCI_User_Passkey_Notification	0x3B	BD_ADDR, Passkey

Description:

The HCI_User_Passkey_Notification event is used to provide a passkey for the Host to display to the user as required as part of a Secure Simple Pairing process. The Passkey parameter shall be a randomly generated number (see [Vol 2] Part H, Section 2) generated by the Controller *mod* 1,000,000.

This event will be generated if the IO capabilities of the local device are DisplayOnly or DisplayYesNo and the IO capabilities of the remote device are KeyboardOnly.

This event shall only be generated if Secure Simple Pairing has been enabled with the HCI_Write_Simple_Pairing_Mode command.

Event parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0xFFFFFFFFXXXX	BD_ADDR of the device involved in the Secure Simple Pairing process.

Passkey:

Size: 4 octets

Value	Parameter Description
0x00000000 to 0x000F423F	Passkey to be displayed. Valid values are decimal 000000 to 999999.



7.7.49 Keypress Notification event

Event	Event Code	Event Parameters
HCI_Keypress_ - Notification	0x3C	BD_ADDR, Notification_Type

Description:

The HCI_Keypress_Notification event is sent to the Host after a passkey notification has been received by the Link Manager on the given BD_ADDR. The Notification_Type parameter may be used by the Host's user interface.

Event parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0XXXXXXXXXXXXX	BD_ADDR of remote device involved in the Secure Simple Pairing process

Notification_Type:

Size: 1 octet

Value	Parameter Description
0	Passkey entry started
1	Passkey digit entered
2	Passkey digit erased
3	Passkey cleared
4	Passkey entry completed
All other values	Reserved for future use



7.7.50 Remote Host Supported Features Notification event

Event	Event Code	Event Parameters
HCI_Remote_Host_Supported_Features_Notification	0x3D	BD_ADDR, Host_Supported_Features

Description:

The HCI_Remote_Host_Supported_Features_Notification event is used to return the LMP extended features page containing the Host features. The BD_ADDR shall be the address of the remote device.

This event shall only be generated after the LMP extended features are read from the remote device during a connection initiated by an HCI_Remote_Name_Request command.

Note: This event is not generated during a connection initiated by the HCI_Create_Connection command.

Event parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0XXXXXXXXXXXXX	BD_ADDR of remote device.

Host_Supported_Features:

Size: 8 octets

Value	Parameter Description
0XXXXXXXXXXXXXXXXX	Bit map of Host Supported Features page of LMP extended features. For more information, see [Vol 2] Part C, Link Manager Protocol Specification .



7.7.51 [This section is no longer used]

7.7.52 [This section is no longer used]

7.7.53 [This section is no longer used]

7.7.54 [This section is no longer used]

7.7.55 [[This section is no longer used]

7.7.56 [This section is no longer used]

7.7.57 [This section is no longer used]

7.7.58 [This section is no longer used]



7.7.59 Number Of Completed Data Blocks event

Event	Event Code	Event Parameters
HCI_Number_Of_Completed_Data_Blocks	0x48	Total_Num_Data_Blocks, Num_Handles, Connection_Handle[i], Num_Completed_Packets[i], Num_Completed_Blocks[i]

Description:

The HCI_Number_Of_Completed_Data_Blocks event is used by the Controller to indicate to the Host how many HCI ACL Data packets have been completed (transmitted or flushed), and how many data block buffers have been freed, for each Connection_Handle since the previous HCI_Number_Of_Completed_Data_Blocks event was sent to the Host. This means that the corresponding buffer space has been freed in the Controller. Based on this information, and the Total_Num_Data_Blocks parameter, the Host can determine for which Handles the following HCI ACL Data packets should be sent to the Controller.

The Host should determine the number of blocks occupied by each ACL data packet by dividing the ACL data packet size by the Data_Block_Length parameter of the HCI_Read_Data_Block_Size command.

The Total_Num_Data_Blocks parameter indicates the total number of buffer blocks available in the Controller. Before any HCI_Number_Of_Completed_Data_Blocks event is received, the value of Total_Num_Data_Blocks from the HCI_Read_Data_Block_Size command is used. This allows the value to be updated at any time, which provides the Controller with some flexibility on its buffer allocation.

If the Controller were permitted to reduce its buffer pool in an arbitrary way then there is a potential race condition, in the case where the Host has just started to transmit a new packet. In order to prevent this race condition, the Total_Num_Data_Blocks parameter shall not indicate a reduction in the pool of blocks greater than the sum of the Num_Completed_Blocks values in this event. If a greater reduction in the block pool is required then the value 0 shall be indicated here. The value 0 has a special meaning and indicates that the Host shall re-issue the HCI_Read_Data_Block_Size command in order to find the new buffer pool size. The Host shall wait for any outstanding TX to complete and shall defer further TX until the HCI_Read_Data_Block_Size command has been issued and completed. The Controller shall reduce its allocation only after the HCI_Read_Data_Block_Size command has been issued and completed. This ensures that the race condition described above is avoided.

**Event parameters:****Total_Num_Data_Blocks:****Size: 2 octets**

Value	Parameter Description
0x0000	The size of the buffer pool may have changed. The Host is requested to issue an HCI_Read_Data_Block_Size command in order to determine the new value of Total_Num_Data_Blocks.
0xFFFF	Total number of data block buffers available in the Controller for the storage of data packets scheduled for transmission. This indicates the existing value is unchanged, or increased, or reduced by up to the sum of the Num_Completed_Blocks values in this command.

Num_Handles:**Size: 1 octet**

Value	Parameter Description
0xFF	The number of Handles and Num_Completed_Packets and Num_Completed_Blocks parameter triples contained in this event. Range: 0 to 255

Connection_Handle[i]:**Size: Num_Handles × 2 octets (12 bits meaningful)**

Value	Parameter Description
0xFFFF	Connection_Handle for a BR/EDR connection. Range: 0x0000 to 0x0EFF

Num_Completed_Packets[i]:**Size: Num_Handles × 2 octets**

Value	Parameter Description
0xFFFF	The number of HCI ACL Data packets that have been completed (transmitted or flushed) for the associated Handle since the previous time that an HCI_Number_Of_Completed_Data_Blocks event provided information about this Handle. Range: 0x0000 to 0xFFFF

Num_Completed_Blocks[i]:**Size: Num_Handles × 2 octets**

Value	Parameter Description
0xFFFF	The number of data blocks that have been freed for the associated Handle since the previous time that an HCI_Number_Of_Completed_Data_Blocks event provided information about this Handle. Range: 0x0000 to 0xFFFF



7.7.60 [This section is no longer used]

7.7.61 [This section is no longer used]

7.7.62 [This section is no longer used]

7.7.63 [This section is no longer used]

7.7.64 [This section is no longer used]



7.7.65 LE Meta event

Description:

The LE Meta event is used to encapsulate all LE Controller specific events. The Event Code of all LE Meta events shall be 0x3E. The Subevent_Code is the first octet of the event parameters. The Subevent_Code shall be set to one of the valid Subevent_Codes from an LE specific event. All other Subevent_Parameters are defined in the LE Controller specific events.

7.7.65.1 LE Connection Complete event

Event	Event Code	Event Parameters
HCI_LE_Connection_Complete	0x3E	Subevent_Code, Status, Connection_Handle, Role, Peer_Address_Type, Peer_Address, Connection_Interval, Peripheral_Latency, Supervision_Timeout, Central_Clock_Accuracy

Description:

The HCI_LE_Connection_Complete event indicates to both of the Hosts forming the connection that a new connection has been created. Upon the creation of the connection a Connection_Handle shall be assigned by the Controller, and passed to the Host in this event. If the connection creation fails this event shall be provided to the Host that had issued the HCI_LE_Create_Connection command.

This event indicates to the Host which issued an HCI_LE_Create_Connection command and received an HCI_Command_Status event if the connection creation failed or was successful.

The Central_Clock_Accuracy parameter is only valid for a Peripheral. On a Central, this parameter shall be set to 0x00.

Note: This event is not sent if the HCI_LE_Enhanced_Connection_Complete event (see [Section 7.7.65.10](#)) is unmasked.

**Event parameters:***Subevent_Code:**Size: 1 octet*

Value	Parameter Description
0x01	Subevent code for the HCI_LE_Connection_Complete event

*Status:**Size: 1 octet*

Value	Parameter Description
0x00	Connection successfully completed.
0x01 to 0xFF	Connection failed to complete. [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

*Connection_Handle:**Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

*Role:**Size: 1 octet*

Value	Parameter Description
0x00	Connection is Central
0x01	Connection is Peripheral
All other values	Reserved for future use

*Peer_Address_Type:**Size: 1 octet*

Value	Parameter Description
0x00	Peer is using a Public Device Address
0x01	Peer is using a Random Device Address
All other values	Reserved for future use

*Peer_Address:**Size: 6 octets*

Value	Parameter Description
0xFFFFFFFFXXXX	Public Device Address or Random Device Address of the peer device

Host Controller Interface Functional Specification**Connection_Interval:****Size: 2 octets**

Value	Parameter Description
0xXXXX	Connection interval used on this connection. Range: 0x0006 to 0x0C80 Time = N * 1.25 ms Time Range: 7.5 ms to 4000 ms.

Peripheral_Latency:**Size: 2 octets**

Value	Parameter Description
0xXXXX	Peripheral latency for the connection in number of connection events. Range: 0x0000 to 0x01F3

Supervision_Timeout:**Size: 2 octets**

Value	Parameter Description
0xXXXX	Connection supervision timeout. Range: 0x000A to 0x0C80 Time = N * 10 ms Time Range: 100 ms to 32 s

Central_Clock_Accuracy:**Size: 1 octet**

Value	Parameter Description
0x00	500 ppm
0x01	250 ppm
0x02	150 ppm
0x03	100 ppm
0x04	75 ppm
0x05	50 ppm
0x06	30 ppm
0x07	20 ppm
All other values	Reserved for future use



7.7.65.2 LE Advertising Report event

Event	Event Code	Event Parameters
HCI_LE_Advertising_Report	0x3E	Subevent_Code, Num_Reports, Event_Type[i], Address_Type[i], Address[i], Data_Length[i], Data[i], RSSI[i]

Description:

The HCI_LE_Advertising_Report event indicates that one or more Bluetooth devices have responded to an active scan or have broadcast advertisements that were received during a passive scan. The Controller may queue these advertising reports and send information from multiple devices in one HCI_LE_Advertising_Report event.

This event shall only be generated if scanning was enabled using the HCI_LE_Set_Scan_Enable command. It only reports advertising events that used legacy advertising PDUs.

Event parameters:

Subevent_Code: *Size: 1 octet*

Value	Parameter Description
0x02	Subevent code for the HCI_LE_Advertising_Report event

Num_Reports: *Size: 1 octet*

Value	Parameter Description
0x01 to 0x19	Number of responses in event.
All other values	Reserved for future use

Event_Type[i]: *Size: Num_Reports × 1 octet*

Value	Parameter Description
0x00	Connectable and scannable undirected advertising (ADV_IND)
0x01	Connectable directed advertising (ADV_DIRECT_IND)
0x02	Scannable undirected advertising (ADV_SCAN_IND)
0x03	Non connectable undirected advertising (ADV_NONCONN_IND)



Value	Parameter Description
0x04	Scan Response (SCAN_RSP)
All other values	Reserved for future use

Address_Type[i]: *Size: Num_Reports × 1 octet*

Value	Parameter Description
0x00	Public Device Address
0x01	Random Device Address
0x02	Public Identity Address (Corresponds to Resolved Private Address)
0x03	Random (static) Identity Address (Corresponds to Resolved Private Address)
All other values	Reserved for future use

Address[i]: *Size: Num_Reports × 6 octets*

Value	Parameter Description
0XXXXXXXXXXXXX	Public Device Address, Random Device Address, Public Identity Address or Random (static) Identity Address of the advertising device.

Data_Length[i]: *Size: Num_Reports × 1 octets*

Value	Parameter Description
0x00 to 0x1F	Length of the Data[i] field for the device which responded.
All other values	Reserved for future use.

Data[i]: *Size: SUM (Data_Length[i]) octets*

Value	Parameter Description
	Data_Length[i] octets of advertising or scan response data formatted as defined in [Vol 3] Part C, Section 11 . Note: Each element of this array has a variable length.

RSSI[i]: *Size: Num_Reports × 1 octet*

Value	Parameter Description
0xXX	Range: -127 to +20 Units: dBm
0x7F	RSSI is not available

**7.7.65.3 LE Connection Update Complete event**

Event	Event Code	Event Parameters
HCI_LE_ - Connection_ - Update_Complete	0x3E	Subevent_Code, Status, Connection_Handle, Connection_Interval, Peripheral_Latency, Supervision_Timeout

Description:

The HCI_LE_Connection_Update_Complete event is used to indicate that the Connection Update procedure has completed.

This event shall be issued if the HCI_LE_Connection_Update command was issued by the Host or if the connection parameters are updated following a request from the peer device. If no parameters are updated following a request from the peer device or the parameters were changed using the Connection Subrate Update procedure, then this event shall not be issued.

If the Status parameter is zero and the connection interval has changed, then the Link Layer must have set the subrating factor to 1 and the continuation number to 0 (see [\[Vol 6\] Part B, Section 5.1.1](#)).

Note: This event can be issued autonomously by the Central's Controller if it decides to change the connection interval based on the range of allowable connection intervals for that connection.

Note: The parameter values returned in this event may be different from the parameter values provided by the Host through the HCI_LE_Connection_Update command ([Section 7.8.18](#)) or the HCI_LE_Remote_Connection_Parameter_Request_Reply command ([Section 7.8.31](#)).

Event parameters:*Subevent_Code:**Size: 1 octet*

Value	Parameter Description
0x03	Subevent code for the HCI_LE_Connection_Update_Complete event

**Status:****Size: 1 octet**

Value	Parameter Description
0x00	HCI_LE_Connection_Update command successfully completed.
0x01 to 0xFF	HCI_LE_Connection_Update command failed to complete. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Connection_Handle:**Size: 2 octets (12 bits meaningful)**

Value	Parameter Description
0XXXXX	Connection_Handle Range: 0x0000 to 0x0EFF

Connection_Interval:**Size: 2 octets**

Value	Parameter Description
0XXXXX	Connection interval used on this connection. Range: 0x0006 to 0x0C80 Time = N * 1.25 ms Time Range: 7.5 ms to 4000 ms.

Peripheral_Latency:**Size: 2 octets**

Value	Parameter Description
0XXXXX	Peripheral latency for the connection in number of subrated connection events. Range: 0x0000 to 0x01F3

Supervision_Timeout:**Size: 2 octets**

Value	Parameter Description
0XXXXX	Supervision timeout for this connection. Range: 0x000A to 0x0C80 Time = N * 10 ms Time Range: 100 ms to 32000 ms



7.7.65.4 LE Read Remote Features Complete event

Event	Event Code	Event Parameters
HCI_LE_Read_Remote_Features_Complete	0x3E	Subevent_Code, Status, Connection_Handle, LE_Features

Description:

The HCI_LE_Read_Remote_Features_Complete event is used to indicate the completion of the process of the Controller obtaining the features used on the connection and the features supported by the remote Bluetooth device specified by the Connection_Handle event parameter.

Note: If the features are requested more than once while a connection exists between the two devices, the second and subsequent requests may report a cached copy of the features rather than fetching the feature mask again.

Event parameters:

Subevent_Code: *Size: 1 octet*

Value	Parameter Description
0x04	Subevent code for the HCI_LE_Read_Remote_Features_Complete event.

Status: *Size: 1 octet*

Value	Parameter Description
0x00	HCI_LE_Read_Remote_Features command successfully completed.
0x01 to 0xFF	HCI_LE_Read_Remote_Features command failed to complete. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Connection_Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

LE_Features: *Size: 8 octets*

Value	Parameter Description
0xFFFFFFFF XXXXXX	Bit Mask List of LE features. See [Vol 6] Part B, Section 4.6 .



7.7.65.5 LE Long Term Key Request event

Event	Event Code	Event Parameters
HCI_LE_Long_Term_Key_Request	0x3E	Subevent_Code, Connection_Handle, Random_Number, Encrypted_Diversifier

Description:

The HCI_LE_Long_Term_Key_Request event indicates that the peer device, in the Central role, is attempting to encrypt or re-encrypt the link and is requesting the Long Term Key from the Host. (See [\[Vol 6\] Part B, Section 5.1.3](#)).

This event shall only be generated when the local device's role is Peripheral.

Event parameters:

Subevent_Code:

Size: 1 octet

Value	Parameter Description
0x05	Subevent code for the HCI_LE_Long_Term_Key_Request event

Connection_Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

Random_Number:

Size: 8 octets

Value	Parameter Description
0xFFFFFFFF XXXXXX	64-bit random number.

Encrypted_Diversifier:

Size: 2 octets

Value	Parameter Description
0xFFFF	16-bit encrypted diversifier.

**7.7.65.6 LE Remote Connection Parameter Request event**

Event	Event Code	Event Parameters
HCI_LE_Remote_Connection_Parameter_Request	0x3E	Subevent_Code, Connection_Handle, Interval_Min, Interval_Max, Max_Latency, Timeout

Description:

This event indicates to the Central's Host or the Peripheral's Host that the remote device is requesting a change in the connection parameters using the Connection Update procedure. The Host replies either with the HCI_LE_Remote_Connection_Parameter_Request_Reply command or the HCI_LE_Remote_Connection_Parameter_Request_Negative_Reply command.

Event parameters:

Subevent_Code: *Size: 1 octet*

Value	Parameter Description
0x06	Subevent code for the HCI_LE_Remote_Connection_Parameter_Request event.

Connection_Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xFFFF	Connection_Handle Range 0x0000 to 0x0EFF

Interval_Min: *Size: 2 octets*

Value	Parameter Description
N = 0xFFFF	Minimum value of the connection interval requested by the remote device. Range: 0x0006 to 0x0C80 Time = N * 1.25 ms Time Range: 7.5 ms to 4 s

**Interval_Max:****Size: 2 octets**

Value	Parameter Description
N = 0xXXXX	Maximum value of the connection interval requested by the remote device. Range: 0x0006 to 0x0C80 Time = N * 1.25 ms Time Range: 7.5 ms to 4 s

Max_Latency:**Size: 2 octets**

Value	Parameter Description
0xXXXX	Maximum allowed Peripheral latency for the connection specified as the number of subrated connection events requested by the remote device. Range: 0x0000 to 0x01F3 (499)

Timeout:**Size: 2 octets**

Value	Parameter Description
N = 0xXXXX	Supervision timeout for the connection requested by the remote device. Range: 0x000A to 0x0C80 Time = N * 10 ms Time Range: 100 ms to 32 s



7.7.65.7 LE Data Length Change event

Event	Event Code	Event Parameters
HCI_LE_Data_Length_Change	0x3E	Subevent_Code, Connection_Handle, Max_TX_Octets, Max_TX_Time, Max_RX_Octets, Max_RX_Time

Description:

The HCI_LE_Data_Length_Change event notifies the Host of a change to either the maximum LL Data PDU Payload length or the maximum transmission time of packets containing LL Data PDUs in either direction. The values reported are the limits imposed on the connection by the Link Layer following the change (see [\[Vol 6\] Part B, Section 4.5.10](#)); the actual maximum used on the connection may be less for other reasons. This event shall not be generated if the values have not changed.

Event parameters:

Subevent_Code:

Size: 1 octet

Value	Parameter Description
0x07	Subevent code for the HCI_LE_Data_Length_Change event

Connection_Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xFFFF	Connection_Handle Range 0x0000 to 0x0EFF

Max_TX_Octets:

Size: 2 octets

Value	Parameter Description
0xFFFF	The maximum number of payload octets in a LLData PDU that the local Controller will send on this connection (<i>connEffectiveMaxTxOctets</i> defined in [Vol 6] Part B, Section 4.5.10). Range 0x001B to 0x00FB

**Max_TX_Time:****Size: 2 octets**

Value	Parameter Description
0xXXXX	The maximum time that the local Controller will take to send a Link Layer packet containing an LL Data PDU on this connection (<i>connEffectiveMaxTxTime</i> defined in [Vol 6] Part B, Section 4.5.10). Range 0x0148 to 0x4290

Max_RX_Octets:**Size: 2 octets**

Value	Parameter Description
0xXXXX	The maximum number of payload octets in a Link Layer packet that the local Controller expects to receive on this connection (<i>connEffectiveMaxRxOctets</i> defined in [Vol 6] Part B, Section 4.5.10). Range 0x001B to 0x00FB

Max_RX_Time:**Size: 2 octets**

Value	Parameter Description
0xXXXX	The maximum time that the local Controller expects to take to receive a Link Layer packet on this connection (<i>connEffectiveMaxRxTime</i> defined in [Vol 6] Part B, Section 4.5.10). Range 0x0148 to 0x4290



7.7.65.8 LE Read Local P-256 Public Key Complete event

Event	Event Code	Event Parameters
HCI_LE_Read_Local_P-256_Public_Key - Complete	0x3E	Subevent_Code, Status, Key_X_Coordinate, Key_Y_Coordinate

Description:

This event is generated when local P-256 key generation is complete.

Event parameters:

Subevent_Code: *Size: 1 octet*

Value	Parameter Description
0x08	Subevent code for the HCI_LE_Read_Local_P-256_Public_Key_Complete event.

Status: *Size: 1 octet*

Value	Parameter Description
0x00	HCI_LE_Read_Local_P-256_Public_Key command completed successfully.
0x01 to 0xFF	HCI_LE_Read_Local_P-256_Public_Key command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Key_X_Coordinate: *Size: 32 octets*

Value	Parameter Description
0xFFFFFFFFFFFFFFFF XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX	Local P-256 public key X coordinate.

Key_Y_Coordinate: *Size: 32 octets*

Value	Parameter Description
0xFFFFFFFFFFFFFFFF XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX	Local P-256 public key Y coordinate.

**7.7.65.9 LE Generate DHKey Complete event**

Event	Event Code	Event Parameters
HCI_LE_Generate_DHKey_Complete	0x3E	Subevent_Code, Status, DH_Key

Description:

This event indicates that LE Diffie Hellman key generation has been completed by the Controller.

If the Remote_P-256_Public_Key parameter of the HCI_LE_Generate_DHKey command (see [Section 7.8.37](#)) was invalid (see [\[Vol 3\] Part H, Section 2.3.5.6.1](#)), then all octets of the DH_Key event parameter should be set to 0xFF.

Event parameters:**Subevent_Code:***Size: 1 octet*

Value	Parameter Description
0x09	Subevent code for the HCI_LE_Generate_DHKey_Complete event.

Status:*Size: 1 octet*

Value	Parameter Description
0x00	HCI_LE_Generate_DHKey command completed successfully.
0x01 to 0xFF	HCI_LE_Generate_DHKey command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

DH_Key:*Size: 32 octets*

Value	Parameter Description
0xFFFFFFFF XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX	Diffie Hellman Key.



7.7.65.10 LE Enhanced Connection Complete event

Event	Event Code	Event Parameters
HCI_LE_Enhanced_Connection_Complete	0x3E	Subevent_Code, Status, Connection_Handle, Role, Peer_Address_Type, Peer_Address, Local_Resolvable_Private_Address, Peer_Resolvable_Private_Address Connection_Interval, Peripheral_Latency, Supervision_Timeout, Central_Clock_Accuracy

Description:

The HCI_LE_Enhanced_Connection_Complete event indicates to both of the Hosts forming the connection that a new connection has been created. Upon the creation of the connection a Connection_Handle shall be assigned by the Controller, and passed to the Host in this event. If the connection creation fails, this event shall be provided to the Host that had issued the HCI_LE_Create_Connection or HCI_LE_Extended_Create_Connection command.

If this event is unmasked and the HCI_LE_Connection_Complete event is unmasked, only the HCI_LE_Enhanced_Connection_Complete event is sent when a new connection has been created.

This event indicates to the Host that issued an HCI_LE_Create_Connection or HCI_LE_Extended_Create_Connection command and received an HCI_Command_Status event if the connection creation failed or was successful.

The Peer_Address, Peer_Resolvable_Private_Address, and Local_Resolvable_Private_Address shall always reflect the most recent packet sent and received on air.

The Central_Clock_Accuracy parameter is only valid for a Peripheral. On a Central, this parameter shall be set to 0x00.

**Event parameters:***Subevent_Code:**Size: 1 octet*

Value	Parameter Description
0x0A	Subevent code for the HCI_LE_Enhanced_Connection_Complete event

*Status:**Size: 1 octet*

Value	Parameter Description
0x00	Connection successfully completed.
0x01 to 0xFF	Connection failed to complete. See [Vol 1] Part F for a list of error codes and descriptions.

*Connection_Handle:**Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

*Role:**Size: 1 octet*

Value	Parameter Description
0x00	Connection is Central
0x01	Connection is Peripheral
All other values	Reserved for future use

*Peer_Address_Type:**Size: 1 octet*

Value	Parameter Description
0x00	Public Device Address (default)
0x01	Random Device Address
0x02	Public Identity Address (Corresponds to Resolved Private Address)
0x03	Random (Static) Identity Address (Corresponds to Resolved Private Address)
All other values	Reserved for future use

*Peer_Address:**Size: 6 octets*

Value	Parameter Description
0XXXXXXXXXX XX	Public Device Address, or Random Device Address, Public Identity Address or Random (static) Identity Address of the device to be connected.

**Local_Resolvable_Private_Address:****Size: 6 octets**

Value	Parameter Description
0XXXXXXXXXX XX	Resolvable Private Address being used by the local device for this connection. This is only valid when the Own_Address_Type (from the HCI_LE_Create_Connection, HCI_LE_Set_Advertising_Parameters, HCI_LE_Set_Extended_Advertising_Parameters, or HCI_LE_Extended_Create_Connection commands) is set to 0x02 or 0x03, and the Controller generated a resolvable private address for the local device using a non-zero local IRK. For other Own_Address_Type values, the Controller shall return all zeros.

Peer_Resolvable_Private_Address:**Size: 6 octets**

Value	Parameter Description
0XXXXXXXXXX XX	Resolvable Private Address being used by the peer device for this connection. This is only valid for Peer_Address_Type 0x02 and 0x03. For other Peer_Address_Type values, the Controller shall return all zeros.

Connection_Interval:**Size: 2 octets**

Value	Parameter Description
N = 0XXXXX	Connection interval used on this connection. Range: 0x0006 to 0x0C80 Time = N * 1.25 ms Time Range: 7.5 ms to 4000 ms.

Peripheral_Latency:**Size: 2 octets**

Value	Parameter Description
0XXXXX	Peripheral latency for the connection in number of connection events. Range: 0x0000 to 0x01F3

Supervision_Timeout:**Size: 2 octets**

Value	Parameter Description
N = 0XXXXX	Connection supervision timeout. Range: 0x000A to 0x0C80 Time = N * 10 ms Time Range: 100 ms to 32 s

Central_Clock_Accuracy:**Size: 1 octet**

Value	Parameter Description
0x00	500 ppm
0x01	250 ppm



Value	Parameter Description
0x02	150 ppm
0x03	100 ppm
0x04	75 ppm
0x05	50 ppm
0x06	30 ppm
0x07	20 ppm
All other values	Reserved for future use

**7.7.65.11 LE Directed Advertising Report event**

Event	Event Code	Event Parameters
HCI_LE_Directed_Advertising_Report	0x3E	Subevent_Code, Num_Reports, Event_Type[i], Address_Type[i], Address[i], Direct_Address_Type[i], Direct_Address[i], RSSI[i]

Description:

The HCI_LE_Directed_Advertising_Report event indicates that directed advertisements have been received where the advertiser is using a resolvable private address for the TargetA field of the advertising PDU which the Controller is unable to resolve and the Scanning_Filter_Policy is equal to 0x02 or 0x03, see [Section 7.8.10](#). Direct_Address_Type and Direct_Address specify the address the directed advertisements are being directed to. Address_Type and Address specify the address of the advertiser sending the directed advertisements. The Controller may queue these advertising reports and send information from multiple advertisers in one HCI_LE_Directed_Advertising_Report event.

This event shall only be generated if scanning was enabled using the HCI_LE_Set_Scan_Enable command. It only reports advertising events that used legacy advertising PDUs.

Event parameters:

Subevent_Code: *Size: 1 octet*

Value	Parameter Description
0x0B	Subevent code for the HCI_LE_Directed_Advertising_Report event

Num_Reports: *Size: 1 octet*

Value	Parameter Description
0x01 to 0x19	Number of responses in event
All other values	Reserved for future use

*Event_Type[i]:**Size: Num_Reports × 1 octet*

Value	Parameter Description
0x01	Connectable directed legacy advertising (ADV_DIRECT_IND)
All other values	Reserved for future use

*Address_Type[i]:**Size: Num_Reports × 1 octet*

Value	Parameter Description
0x00	Public Device Address (default)
0x01	Random Device Address
0x02	Public Identity Address (Corresponds to Resolved Private Address)
0x03	Random (static) Identity Address (Corresponds to Resolved Private Address)
All other values	Reserved for future use

*Address[i]:**Size: Num_Reports × 6 octets*

Value	Parameter Description
0XXXXXXXXXX XX	Public Device Address, Random Device Address, Public Identity Address or Random (static) Identity Address of the advertising device.

*Direct_Address_Type[i]:**Size: Num_Reports × 1 octet*

Value	Parameter Description
0x01	Random Device Address (default)
All other values	Reserved for future use

*Direct_Address[i]:**Size: Num_Reports × 6 octets*

Value	Parameter Description
0XXXXXXXXXX XX	Random Device Address

*RSSI[i]:**Size: Num_Reports × 1 octet*

Value	Parameter Description
0xXX	Range: -127 to +20 Units: dBm
0x7F	RSSI is not available



7.7.65.12 LE PHY Update Complete event

Event	Event Code	Event Parameters
HCI_LE_PHY_Update_Complete	0x3E	Subevent_Code, Status, Connection_Handle, TX_PHY, RX_PHY

Description:

The HCI_LE_PHY_Update_Complete event is used to indicate that the Controller has changed the transmitter PHY or receiver PHY in use.

If the Controller changes the transmitter PHY, the receiver PHY, or both PHYs, this event shall be issued.

If an HCI_LE_Set_PHY command was sent and the Controller determines that neither PHY will change as a result, it issues this event immediately.

Event parameters:

Subevent_Code: *Size: 1 octet*

Value	Parameter Description
0x0C	Subevent code for the HCI_LE_PHY_Update_Complete event

Status: *Size: 1 octet*

Value	Parameter Description
0x00	HCI_LE_Set_PHY command succeeded or autonomous PHY update made by the Controller.
0x01 to 0xFF	HCI_LE_Set_PHY command failed. See [Vol 1] Part F for a list of error codes and descriptions.

Connection_Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

TX_PHY: *Size: 1 octet*

Value	Parameter Description
0x01	The transmitter PHY for the connection is LE 1M
0x02	The transmitter PHY for the connection is LE 2M



Value	Parameter Description
0x03	The transmitter PHY for the connection is LE Coded
All other values	Reserved for future use

*RX_PHY:**Size: 1 octet*

Value	Parameter Description
0x01	The receiver PHY for the connection is LE 1M
0x02	The receiver PHY for the connection is LE 2M
0x03	The receiver PHY for the connection is LE Coded
All other values	Reserved for future use



7.7.65.13 LE Extended Advertising Report event

Event	Event Code	Event Parameters
HCI_LE_Extended_Advertising_Report	0x3E	Subevent_Code, Num_Reports, Event_Type[i], Address_Type[i], Address[i], Primary_PHY[i], Secondary_PHY[i], Advertising_SID[i], TX_Power[i], RSSI[i], Periodic_Advertising_Interval[i], Direct_Address_Type[i], Direct_Address[i], Data_Length[i], Data[i]

Description:

The HCI_LE_Extended_Advertising_Report event indicates that one or more Bluetooth devices have responded to an active scan or have broadcast advertisements that were received during a passive scan. The Controller may coalesce multiple advertising reports from the same or different advertisers into a single HCI_LE_Extended_Advertising_Report event, provided all the parameters from all the advertising reports fit in a single HCI event.

This event shall only be generated if scanning was enabled using the HCI_LE_Set_Extended_Scan_Enable command. It reports advertising events using either legacy or extended advertising PDUs.

The Controller may split the data from a single advertisement or scan response (whether one PDU or several) into several reports. If so, each report except the last shall have an Event_Type with a data status field of "incomplete, more data to come", while the last shall have the value "complete"; the Address_Type, Address, Advertising_SID, Primary_PHY, and Secondary_PHY fields shall be the same in all the reports. No further reports shall be sent for a given advertisement or scan response after one with a Data_Status other than "incomplete, more data to come".

When a scan response is received, bits 0 to 2 and 4 of the event type shall indicate the properties of the original advertising event and the Advertising_SID field should be set to the value in the original scannable advertisement.



An Event_Type with a data status field of "incomplete, data truncated" indicates that the Controller attempted to receive an AUX_CHAIN_IND PDU but was not successful.

Where the event being reported used a legacy advertising PDU, the Controller shall set the Event_Type to the value specified in [Table 7.1](#).

PDU Type	Event_Type
ADV_IND	0b0010011
ADV_DIRECT_IND	0b0010101
ADV_SCAN_IND	0b0010010
ADV_NONCONN_IND	0b0010000
SCAN_RSP to an ADV_IND	0b0011011
SCAN_RSP to an ADV_SCAN_IND	0b0011010

Table 7.1: Event_Type values for legacy PDUs

If the Event_Type indicates a legacy PDU (bit 4 = 1), the Primary_PHY parameter shall indicate the LE 1M PHY and the Secondary_PHY parameter shall be set to 0x00. Otherwise, the Primary_PHY parameter indicates the PHY used to send the advertising PDU on the primary advertising physical channel and the Secondary_PHY parameter indicates the PHY used to send the advertising PDU(s), if any, on the secondary advertising physical channel.

The Periodic_Advertising_Interval event parameter is set to zero when no periodic advertising exists as part of the advertising set.

The Direct_Address_Type and Direct_Address parameters indicate the TargetA address in the advertising PDU for directed advertising event types (bit 2 = 1). These parameters shall be ignored for undirected advertising event types (bit 2 = 0). If the TargetA address is a resolvable private address that the Controller successfully resolved, then the value of Direct_Address_Type shall depend on the value of the Own_Address_Type parameter of the command that set the extended scan parameters. Direct_Address shall be set as follows:

- If Direct_Address_Type equals 0x02, then Direct_Address shall be set to either the TargetA field in the received advertisement or to the public device address of the scanning device.
- If Direct_Address_Type equals 0x03, then Direct_Address shall be set to either the TargetA field in the received advertisement or to the address set by the HCI_LE_Set_Random_Address command.
- Otherwise Direct_Address shall be set to the TargetA field in the received advertisement.

When multiple advertising packets are used to complete a single advertising report (e.g., a packet containing an ADV_EXT_IND PDU combined with one containing an AUX_ADV_IND PDU), the RSSI event parameter shall be set



based on the last packet received and the TX_Power event parameter shall be based on the last packet of the current advertisement or scan response received that contains a TxPower field. However, if an event has been sent with a TX_Power value other than 0x7F and a Data_Status of "incomplete, more data to come", and if no subsequent PDU with a TxPower field has been received, then subsequent events may instead have a TX_Power value of 0x7F.

If the Controller receives an AUX_CHAIN_IND with no AdvData, it should send the report (or the last report if it has split the data) immediately without waiting for any subsequent AUX_CHAIN_IND PDUs.

Event parameters:

Subevent_Code: *Size: 1 octet*

Value	Parameter Description
0x0D	Subevent code for the HCI_LE_Extended_Advertising_Report event

Num_Reports: *Size: 1 octet*

Value	Parameter Description
0x01 to 0x0A	Number of separate reports in the event
All other values	Reserved for future use

Event_Type[i]: *Size: Num_Reports × 2 octets*

Bit Number	Parameter Description
0	Connectable advertising
1	Scannable advertising
2	Directed advertising
3	Scan response
4	Legacy advertising PDUs used
5 to 6	Data status: 0b00 = Complete 0b01 = Incomplete, more data to come 0b10 = Incomplete, data truncated, no more to come 0b11 = Reserved for future use
All other bits	Reserved for future use

*Address_Type[i]:**Size: Num_Reports × 1 octet*

Value	Parameter Description
0x00	Public Device Address
0x01	Random Device Address
0x02	Public Identity Address (corresponds to Resolved Private Address)
0x03	Random (static) Identity Address (corresponds to Resolved Private Address)
0xFF	No address provided (anonymous advertisement)
All other values	Reserved for future use

*Address[i]:**Size: Num_Reports × 6 octets*

Value	Parameter Description
0XXXXXXXXXXXXX	Public Device Address, Random Device Address, Public Identity Address or Random (static) Identity Address of the advertising device.

*Primary_PHY[i]:**Size: Num_Reports × 1 octet*

Value	Parameter Description
0x01	Advertiser PHY is LE 1M
0x03	Advertiser PHY is LE Coded
All other values	Reserved for future use

*Secondary_PHY[i]:**Size: Num_Reports × 1 octet*

Value	Parameter Description
0x00	No packets on the secondary advertising physical channel
0x01	Advertiser PHY is LE 1M
0x02	Advertiser PHY is LE 2M
0x03	Advertiser PHY is LE Coded
All other values	Reserved for future use

*Advertising_SID[i]:**Size: Num_Reports × 1 octet*

Value	Parameter Description
0x00 to 0x0F	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 advertisement
0xFF	No ADI field provided
All other values	Reserved for future use

***TX_Power[i]:******Size: Num_Reports × 1 octet***

Value	Parameter Description
0xXX	Range: -127 to +20 Units: dBm
0x7F	Tx Power information not available

RSSI[i]:***Size: Num_Reports × 1 octet***

Value	Parameter Description
0xXX	Range: -127 to +20 Units: dBm
0x7F	RSSI is not available

Periodic_Advertising_Interval[i]:***Size: Num_Reports × 2 octets***

Value	Parameter Description
0x0000	No periodic advertising
N = 0xXXXX	Interval of the periodic advertising Range: 0x0006 to 0xFFFF Time = N * 1.25 ms Time Range: 7.5 ms to 81,918.75 s

Direct_Address_Type[i]:***Size: Num_Reports × 1 octet***

Value	Parameter Description
0x00	Public Device Address
0x01	Non-resolvable Private Address or Static Device Address
0x02	Resolvable Private Address (resolved by Controller; Own_Address_Type was 0x00 or 0x02)
0x03	Resolvable Private Address (resolved by Controller; Own_Address_Type was 0x01 or 0x03)
0xFE	Resolvable Private Address (Controller unable to resolve)
All other values	Reserved for future use

Direct_Address[i]:***Size: Num_Reports × 6 octets***

Value	Parameter Description
0XXXXXXXXXXXXX	TargetA field in the advertisement or either Public Identity Address or Random (static) Identity Address of the target device



Data_Length[i]:

Size: Num_Reports × 1 octet

Value	Parameter Description
0 to 229	Length of the Data[i] field for each device which responded
All other values	Reserved for future use

Data[i]:

Size: SUM (Data_Length[i]) octets

Value	Parameter Description
	Data_Length[i] octets of advertising or scan response data formatted as defined in [Vol 3] Part C, Section 11 . Note: Each element of this array has a variable length.

**7.7.65.14 LE Periodic Advertising Sync Established event**

Event	Event Code	Event Parameters
HCI_LE_Periodic_Advertising_Sync_Established	0x3E	Subevent_Code, Status, Sync_Handle, Advertising_SID, Advertiser_Address_Type, Advertiser_Address, Advertiser_PHY, Periodic_Advertising_Interval, Advertiser_Clock_Accuracy

Description:

The HCI_LE_Periodic_Advertising_Sync_Established event indicates that the Controller has received the first periodic advertising packet from an advertiser after the HCI_LE_Periodic_Advertising_Create_Sync command has been sent to the Controller.

The Sync_Handle parameter identifies the periodic advertising train in subsequent commands and events and shall be assigned by the Controller.

The Advertising_SID parameter is set to the value of the Advertising SID subfield in the ADI field of the advertising PDU referring to the periodic advertising train.

The Advertiser_Address_Type and Advertiser_Address parameters specify the address of the periodic advertiser.

The Advertiser_PHY parameter specifies the PHY used for the periodic advertising.

The Periodic_Advertising_Interval parameter specifies the interval between the periodic advertising events.

The Advertiser_Clock_Accuracy parameter specifies the accuracy of the periodic advertiser's clock.

Event parameters:*Subevent_Code:**Size: 1 octet*

Value	Parameter Description
0x0E	Subevent code for the HCI_LE_Periodic_Advertising_Sync_Established event

**Status:****Size: 1 octet**

Value	Parameter Description
0x00	Periodic advertising sync successful
0x01 to 0xFF	Periodic advertising sync failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions

Sync_Handle:**Size: 2 octets (12 bits meaningful)**

Value	Parameter Description
0xFFFF	Sync_Handle identifying the periodic advertising train Range: 0x0000 to 0x0EFF

Advertising_SID:**Size: 1 octet**

Value	Parameter Description
0x00 to 0x0F	Value of the Advertising SID subfield in the ADI field of the PDU

Advertiser_Address_Type:**Size: 1 octet**

Value	Parameter Description
0x00	Public Device Address or Public Identity Address
0x01	Random Device Address or Random (static) Identity Address
0x02	Public Identity Address (corresponds to Resolved Private Address)
0x03	Random (static) Identity Address (corresponds to Resolved Private Address)
All other values	Reserved for future use

Advertiser_Address:**Size: 6 octets**

Value	Parameter Description
0xFFFFFFFFFFFF	Public Device Address, Random Device Address, Public Identity Address, or Random (static) Identity Address of the advertiser

Advertiser_PHY:**Size: 1 octet**

Value	Parameter Description
0x01	Advertiser PHY is LE 1M
0x02	Advertiser PHY is LE 2M
0x03	Advertiser PHY is LE Coded
All other values	Reserved for future use

Host Controller Interface Functional Specification***Periodic_Advertising_Interval:******Size: 2 octets***

Value	Parameter Description
N = 0xFFFF	Periodic advertising interval Range: 0x0006 to 0xFFFF Time = N * 1.25 ms Time Range: 7.5 ms to 81.91875 s

Advertiser_Clock_Accuracy:***Size: 1 octet***

Value	Parameter Description
0x00	500 ppm
0x01	250 ppm
0x02	150 ppm
0x03	100 ppm
0x04	75 ppm
0x05	50 ppm
0x06	30 ppm
0x07	20 ppm
All other values	Reserved for future use



7.7.65.15 LE Periodic Advertising Report event

Event	Event Code	Event Parameters
HCI_LE_Periodic_Advertising_Report	0x3E	Subevent_Code, Sync_Handle, TX_Power, RSSI, CTE_Type, Data_Status, Data_Length, Data

Description:

The HCI_LE_Periodic_Advertising_Report event indicates that the Controller has received a periodic advertisement.

The Sync_Handle parameter identifies the periodic advertising train that the report relates to.

The RSSI parameter contains the RSSI value, excluding any Constant Tone Extension. If the Controller supports the Connectionless CTE Receiver feature, RSSI shall not be set to 0x7F. When multiple advertising packets are used to complete a periodic advertising report (e.g., a packet containing an AUX_SYNC_IND PDU combined with one containing an AUX_CHAIN PDU), the RSSI event parameter shall be set based on the last packet received and the TX_Power event parameter shall be set based on the AUX_SYNC_IND PDU. However, the second or subsequent events for the same periodic advertisement may instead have a TX_Power value of 0x7F.

The Controller may split the data from a single periodic advertisement (whether one PDU or several) into several reports. If so, each report except the last shall have a Data_Status of "incomplete, more data to come", while the last shall have the value "complete". No further reports shall be sent for a given periodic advertisement after one with a Data_Status other than "incomplete, more data to come".

A Data_Status of "incomplete, data truncated" indicates that the Controller attempted to receive an AUX_CHAIN_IND PDU but was not successful.

The CTE_Type parameter indicates the type of Constant Tone Extension in the periodic advertising packets.

If the Controller receives an AUX_CHAIN_IND PDU with no AdvData, it should send the report (or the last report if it has split the data) immediately without waiting for any subsequent AUX_CHAIN_IND PDUs.

**Event parameters:***Subevent_Code:**Size: 1 octet*

Value	Parameter Description
0x0F	Subevent code for the HCI_LE_Periodic_Advertising_Report event

*Sync_Handle:**Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xFFFF	Sync_Handle identifying the periodic advertising train. Range: 0x0000 to 0xFFFF

*TX_Power:**Size: 1 octet*

Value	Parameter Description
0xFF	Range: -127 to +20 Units: dBm
0x7F	Tx Power information not available

*RSSI:**Size: 1 octet*

Value	Parameter Description
0xFF	Range: -127 to +20 Units: dBm
0x7F	RSSI is not available

*CTE_Type:**Size: 1 octet*

Value	Parameter Description
0x00	AoA Constant Tone Extension
0x01	AoD Constant Tone Extension with 1 μ s slots
0x02	AoD Constant Tone Extension with 2 μ s slots
0xFF	No Constant Tone Extension
All other values	Reserved for future use

*Data_Status:**Size: 1 octet*

Value	Parameter Description
0x00	Data complete
0x01	Data incomplete, more data to come



Value	Parameter Description
0x02	Data incomplete, data truncated, no more to come
All other values	Reserved for future use

*Data_Length:**Size: 1 octet*

Value	Parameter Description
0 to 247	Length of the Data field
All other values	Reserved for future use

*Data:**Size: Data_Length octets*

Value	Parameter Description
Variable	Data received from a Periodic Advertising packet

**7.7.65.16 LE Periodic Advertising Sync Lost event**

Event	Event Code	Event Parameters
HCI_LE_Periodic_Advertising_Sync_Lost	0x3E	Subevent_Code, Sync_Handle

Description:

The HCI_LE_Periodic_Advertising_Sync_Lost event indicates that the Controller has not received a Periodic Advertising packet from the train identified by Sync_Handle within the timeout period.

Event parameters:*Subevent_Code:**Size: 1 octet*

Value	Parameter Description
0x10	Subevent code for the HCI_LE_Periodic_Advertising_Sync_Lost event

*Sync_Handle:**Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xFFFF	Sync_Handle identifying the periodic advertising train. Range: 0x0000 to 0x0EFF

**7.7.65.17 LE Scan Timeout event**

Event	Event Code	Event Parameters
HCI_LE_Scan_Timeout	0x3E	Subevent_Code

Description:

The HCI_LE_Scan_Timeout event indicates that scanning has ended because the duration has expired.

This event shall only be generated if scanning was enabled using the HCI_LE_Set_Extended_Scan_Enable command.

Event parameters:

Subevent_Code:

Size: 1 octet

Value	Parameter Description
0x11	Subevent code for the HCI_LE_Scan_Timeout event

**7.7.65.18 LE Advertising Set Terminated event**

Event	Event Code	Event Parameters
HCI_LE_- Advertising_Set_- Terminated	0x3E	Subevent_Code, Status, Advertising_Handle, Connection_Handle, Num_Completed_Extended_Advertising_Events

Description:

The HCI_LE_Advertising_Set_Terminated event indicates that the Controller has terminated advertising in the advertising sets specified by the Advertising_Handle parameter.

This event shall be generated every time connectable advertising in an advertising set results in a connection being created or because the advertising duration or the maximum number of extended advertising events has been reached. It shall not be generated if the Host disables the advertising set.

This event shall only be generated if advertising was enabled using the HCI_LE_Set_Extended_Advertising_Enable command.

The Connection_Handle parameter is only valid when advertising ends because a connection was created.

If the Max_Extended_Advertising_Events parameter in the HCI_LE_Set_Extended_Advertising_Enable command was non-zero, the Num_Completed_Extended_Advertising_Events parameter shall be set to the number of completed extended advertising events the Controller had transmitted when either the duration elapsed or the maximum number of extended advertising events was reached; otherwise it shall be set to zero.

If advertising has terminated as a result of the advertising duration elapsing, the Status parameter shall be set to the error code *Advertising Timeout* (0x3C).

If advertising has terminated because the Max_Extended_Advertising_Events was reached, the Status parameter shall be set to the error code *Limit Reached* (0x43).

Event parameters:*Subevent_Code:**Size: 1 octet*

Value	Parameter Description
0x12	Subevent code for the HCI_LE_Advertising_Set_Terminated event

**Status:****Size: 1 octet**

Value	Parameter Description
0x00	Advertising successfully ended with a connection being created
0x01 to 0xFF	Advertising ended for another reason. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Advertising_Handle:**Size: 1 octet**

Value	Parameter Description
0xXX	Advertising_Handle in which advertising has ended Range: 0x00 to 0xEF

Connection_Handle:**Size: 2 octets (12 bits meaningful)**

Value	Parameter Description
0xFFFF	Connection_Handle of the connection whose creation ended the advertising Range: 0x0000 to 0xFEFF

Num_Completed_Extended_Advertising_Events:**Size: 1 octet**

Value	Parameter Description
0xXX	Number of completed extended advertising events transmitted by the Controller



7.7.65.19 LE Scan Request Received event

Event	Event Code	Event Parameters
HCI_LE_Scan_Request_Received	0x3E	Subevent_Code, Advertising_Handle, Scanner_Address_Type, Scanner_Address

Description:

The HCI_LE_Scan_Request_Received 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 that is allowed by the advertising filter policy. The advertising set is identified by Advertising_Handle.

This event shall only be generated if advertising was enabled using the HCI_LE_Set_Extended_Advertising_Enable command.

The Scanner_Address_Type and Scanner_Address indicates the type of the address and the address of the scanner device.

Event parameters:

Subevent_Code: *Size: 1 octet*

Value	Parameter Description
0x13	Subevent code for the HCI_LE_Scan_Request_Received event

Advertising_Handle: *Size: 1 octet*

Value	Parameter Description
0xXX	Used to identify an advertising set Range: 0x00 to 0xEF

Scanner_Address_Type: *Size: 1 octet*

Value	Parameter Description
0x00	Public Device Address
0x01	Random Device Address
0x02	Public Identity Address (corresponds to Resolved Private Address)
0x03	Random (static) Identity Address (corresponds to Resolved Private Address)
All other values	Reserved for future use



Scanner_Address:

Size: 6 octets

Value	Parameter Description
0XXXXXXXXXXXXX	Public Device Address, Random Device Address, Public Identity Address or Random (static) Identity Address of the advertising device

**7.7.65.20 LE Channel Selection Algorithm event**

Event	Event Code	Event Parameters
HCI_LE_Channel_Selection_Algorithm	0x3E	Subevent_Code, Connection_Handle, Channel_Selection_Algorithm

Description:

The HCI_LE_Channel_Selection_Algorithm event indicates which channel selection algorithm is used on a data physical channel connection (see [\[Vol 6\] Part B, Section 4.5.8](#)).

Event parameters:

Subevent_Code: *Size: 1 octet*

Value	Parameter Description
0x14	Subevent code for the HCI_LE_Channel_Selection_Algorithm event

Connection_Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

Channel_Selection_Algorithm: *Size: 1 octet*

Value	Parameter Description
0x00	LE Channel Selection Algorithm #1 is used
0x01	LE Channel Selection Algorithm #2 is used
All other values	Reserved for future use



7.7.65.21 LE Connectionless IQ Report event

Event	Event Code	Event Parameters
HCI_LE_Connectionless_IQ_Report	0x3E	Subevent_Code, Sync_Handle, Channel_Index, RSSI, RSSI_Antenna_ID, CTE_Type, Slot_Durations, Packet_Status, Periodic_Event_Counter, Sample_Count, I_Sample[i], Q_Sample[i]

Description:

The HCI_LE_Connectionless_IQ_Report event is used by the Controller to report IQ information from the Constant Tone Extension of a received advertising packet forming part of the periodic advertising train identified by Sync_Handle and to report IQ information from the Constant Tone Extension of a received Test Mode packet (see [Section 7.8.28](#)).

The index of the channel on which the packet was received, the RSSI of the packet (excluding the Constant Tone Extension), the ID of the antenna on which this was measured, the type of Constant Tone Extension, the value of *paEventCounter*, and the IQ samples of the Constant Tone Extension of the advertisement are reported in the corresponding parameters. For any given sample, either both or neither of I_Sample[i] and Q_Sample[i] shall equal 0x80.

The Slot_Durations parameter specifies the sampling rate used by the Controller.

The Packet_Status parameter indicates whether the received packet had a valid CRC and, if not, whether the Controller has determined the position and size of the Constant Tone Extension using the Length and CTETime fields.

Note: A Controller is not required to generate this event for packets that have a bad CRC.

The Constant Tone Extension format is defined in [\[Vol 6\] Part B, Section 2.5.1](#).

If the PDU contains AdvData, then any HCI_LE_Periodic_Advertising_Report event triggered by this PDU shall be generated before this event.



Not all PDUs in a periodic advertisement will necessarily trigger an HCI_LE_Periodic_Advertising_Report event. For example, PDUs without AdvData might not trigger that event (see [Vol 6] Part B, Section 4.4.5.1).

The Controller is not required to generate this event for a Constant Tone Extension with a type that it does not support.

This event is also used by the Controller to report that it has insufficient resources to report IQ samples for all received Constant Tone Extensions and has failed to sample at least once. In this case Packet_Status shall be set to 0xFF and Sample_Count to 0x00.

Event parameters:

Subevent_Code: *Size: 1 octet*

Value	Parameter Description
0x15	Subevent code for HCI_LE_Connectionless_IQ_Report event

Sync_Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xFFFF	Sync_Handle identifying the periodic advertising train. Range: 0x0000 to 0x0EFF
0x0FFF	Receiver Test

Channel_Index: *Size: 1 octet*

Value	Parameter Description
0x00 to 0x27	The index of the channel on which the packet was received. Note: 0x25 to 0x27 can be used only for packets generated during test modes.
All other values	Reserved for future use

RSSI: *Size: 2 octets*

Value	Parameter Description
0xFFFF	RSSI of the packet Range: -1270 to +200 Units: 0.1 dBm

RSSI_Antenna_ID: *Size: 1 octet*

Value	Parameter Description
0xFF	Antenna ID

*CTE_Type:**Size: 1 octet*

Value	Parameter Description
0x00	AoA Constant Tone Extension
0x01	AoD Constant Tone Extension with 1 μ s slots
0x02	AoD Constant Tone Extension with 2 μ s slots
All other values	Reserved for future use

*Slot_Durations:**Size: 1 octet*

Value	Parameter Description
0x01	Switching and sampling slots are 1 μ s each
0x02	Switching and sampling slots are 2 μ s each
All other values	Reserved for future use

*Packet_Status:**Size: 1 octet*

Value	Parameter Description
0x00	CRC was correct
0x01	CRC was incorrect and the Length and CTETime fields of the packet were used to determine sampling points
0x02	CRC was incorrect but the Controller has determined the position and length of the Constant Tone Extension in some other way
0xFF	Insufficient resources to sample (Channel_Index, CTE_Type, and Slot_Durations invalid).
All other values	Reserved for future use

*Periodic_Event_Counter:**Size: 2 octets*

Value	Parameter Description
0xFFFF	The value of <i>paEventCounter</i> (see [Vol 6] Part B, Section 4.4.2.1) for the reported AUX_SYNC_IND PDU

*Sample_Count:**Size: 1 octet*

Value	Parameter Description
0x00	No samples provided (only permitted if Packet_Status is 0xFF).
0x09 to 0x52	Total number of sample pairs (there shall be the same number of I samples and Q samples). Note: This number is dependent on the switch and sample slot durations used.
All other values	Reserved for future use

*I_Sample[i]:**Size: Sample_Count × 1 octet*

Value	Parameter Description
0x80	No valid sample available
All other values	I sample for the reported packet (signed integer). The list is in the order of the sampling points within the packet.

*Q_Sample[i]:**Size: Sample_Count × 1 octet*

Value	Parameter Description
0x80	No valid sample available
All other values	Q sample for the reported packet (signed integer). The list is in the order of the sampling points within the packet.



7.7.65.22 LE Connection IQ Report event

Event	Event Code	Event Parameters
HCI_LE_Connection_IQ_Report	0x3E	Subevent_Code, Connection_Handle, RX_PHY, Data_Channel_Index, RSSI, RSSI_Antenna_ID, CTE_Type, Slot_Durations, Packet_Status, Connection_Event_Counter, Sample_Count, I_Sample[i], Q_Sample[i]

Description:

The HCI_LE_Connection_IQ_Report event is used by the Controller to report the IQ samples from the Constant Tone Extension of a received packet (see [\[Vol 6\] Part B, Section 2.4](#)).

The Connection_Handle parameter identifies the connection that corresponds to the reported information.

The receiver PHY, the index of the data channel, the RSSI value of the packet (excluding the Constant Tone Extension), the ID of the antenna on which this was measured, the type of Constant Tone Extension, the value of *connEventCounter*, and the IQ samples of the Constant Tone Extension of the received packet are reported in the corresponding parameters. For any given sample, either both or neither of I_Sample[i] and Q_Sample[i] shall equal 0x80.

The Slot_Durations parameter specifies the sampling rate used by the Controller.

The Packet_Status parameter indicates whether the received packet had a valid CRC and, if not, whether the Controller has determined the position and size of the Constant Tone Extension using the Length and CTETime fields.

Note: A Controller is not required to generate this event for packets that have a bad CRC.

This event is also used by the Controller to report that it has insufficient resources to report IQ samples for all received Constant Tone Extensions and has failed to sample at least once. In this case Packet_Status shall be set to 0xFF and Sample_Count to 0x00.



The Constant Tone Extension format is defined in [\[Vol 6\] Part B, Section 2.1.5](#).

Event parameters:

Subevent_Code: *Size: 1 octet*

Value	Parameter Description
0x16	Subevent code for HCI_LE_Connection_IQ_Report event

Connection_Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

RX_PHY: *Size: 1 octet*

Value	Parameter Description
0x01	The receiver PHY for the connection is LE 1M
0x02	The receiver PHY for the connection is LE 2M
All other values	Reserved for future use

Data_Channel_Index: *Size: 1 octet*

Value	Parameter Description
0x00 to 0x24	The index of the data channel on which the Data Physical Channel PDU was received.
All other values	Reserved for future use

RSSI: *Size: 2 octets*

Value	Parameter Description
0xFFFF	RSSI of the packet Range: -1270 to +200 Units: 0.1 dBm

RSSI_Antenna_ID: *Size: 1 octet*

Value	Parameter Description
0xFF	ID of the antenna on which the RSSI is measured

**CTE_Type:****Size: 1 octet**

Value	Parameter Description
0x00	AoA Constant Tone Extension
0x01	AoD Constant Tone Extension with 1 μ s slots
0x02	AoD Constant Tone Extension with 2 μ s slots
All other values	Reserved for future use

Slot_Durations:**Size: 1 octet**

Value	Parameter Description
0x01	Switching and sampling slots are 1 μ s each
0x02	Switching and sampling slots are 2 μ s each
All other values	Reserved for future use

Packet_Status:**Size: 1 octet**

Value	Parameter Description
0x00	CRC was correct
0x01	CRC was incorrect and the Length and CTETime fields of the packet were used to determine sampling points
0x02	CRC was incorrect but the Controller has determined the position and length of the Constant Tone Extension in some other way
0xFF	Insufficient resources to sample (Data_Channel_Index, CTE_Type, and Slot_Durations invalid).
All other values	Reserved for future use

Connection_Event_Counter:**Size: 2 octets**

Value	Parameter Description
0xFFFF	The value of <i>connEventCounter</i> (see [Vol 6] Part B, Section 4.5.1) for the reported PDU

Sample_Count:**Size: 1 octet**

Value	Parameter Description
0x00	No samples provided (only permitted if Packet_Status is 0xFF).
0x09 to 0x52	Total number of sample pairs (there shall be the same number of I samples and Q samples). Note: This number is dependent on the switch and sample slot durations used.
All other values	Reserved for future use

*I_Sample[i]:**Size: Sample_Count × 1 octet*

Value	Parameter Description
0x80	No valid sample available
All other values	I sample for the reported PDU (signed integer). The list is in the order of the sampling points within the PDU.

*Q_Sample[i]:**Size: Sample_Count × 1 octet*

Value	Parameter Description
0x80	No valid sample available
All other values	Q sample for the reported PDU (signed integer). The list is in the order of the sampling points within the PDU.

**7.7.65.23 LE CTE Request Failed event**

Event	Event Code	Event Parameters
HCI_LE_CTE_Request_Failed	0x3E	Subevent_Code, Status, Connection_Handle

Description:

This event is used by the Controller to report an issue following a request to a peer device to reply with a packet containing an LL_CTE_RSP PDU and a Constant Tone Extension. It shall be generated if the packet containing the LL_CTE_RSP PDU sent in response did not contain a Constant Tone Extension or if the peer rejected the request. It shall not be generated if the packet containing the LL_CTE_RSP PDU had a CRC error or if the procedure response timeout timer (see [\[Vol 6\] Part B, Section 5.2](#)) expired.

Event parameters:**Subevent_Code:****Size: 1 octet**

Value	Parameter Description
0x17	Subevent code for HCI_LE_CTE_Request_Failed event

Status:**Size: 1 octet**

Value	Parameter Description
0x00	LL_CTE_RSP PDU received successfully but without a Constant Tone Extension field.
0x01 to 0xFF	Peer rejected the request. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Connection_Handle:**Size: 2 octets (12 bits meaningful)**

Value	Parameter Description
0xFFFF	Connection_Handle Range 0x0000 to 0x0EFF

**7.7.65.24 LE Periodic Advertising Sync Transfer Received event**

Event	Event Code	Event Parameters
HCI_LE_Periodic_Advertising_Sync_Transfer_Received	0x3E	Subevent_Code, Status, Connection_Handle, Service_Data, Sync_Handle, Advertising_SID, Advertiser_Address_Type, Advertiser_Address, Advertiser_PHY, Periodic_Advertising_Interval, Advertiser_Clock_Accuracy

Description:

The HCI_LE_Periodic_Advertising_Sync_Transfer_Received event is used by the Controller to report that it has received periodic advertising synchronization information from the device referred to by the Connection_Handle parameter and either successfully synchronized to the periodic advertising train or timed out while attempting to synchronize. The Status will be zero if it successfully synchronized and non-zero otherwise.

The Service_Data value is provided by the Host of the device sending the information.

The Sync_Handle identifies the periodic advertising in subsequent commands and events and shall be assigned by the Controller.

The remaining parameters provide information about the periodic advertising (see [Section 7.7.65.14](#)).

If Status is non-zero, all parameter values are valid except Sync_Handle, which the Host shall ignore.

Note: If the Controller is already synchronized to the periodic advertising train described in the received information, no event will be generated.

Event parameters:*Subevent_Code:**Size: 1 octet*

Value	Parameter Description
0x18	Subevent code for the HCI_LE_Periodic_Advertising_Sync_Transfer_Received event

**Status:****Size: 1 octet**

Value	Parameter Description
0x00	Synchronization to the periodic advertising train succeeded.
0x01 to 0xFF	Synchronization to the periodic advertising train failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Connection_Handle:**Size: 2 octets (12 bits meaningful)**

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

Service_Data:**Size: 2 octets**

Value	Parameter Description
0xFFFF	A value provided by the peer device

Sync_Handle:**Size: 2 octets (12 bits meaningful)**

Value	Parameter Description
0xFFFF	Sync_Handle identifying the periodic advertising train. Range: 0x0000 to 0x0EFF

Advertising_SID:**Size: 1 octet**

Value	Parameter Description
0x00 to 0x0F	Value of the Advertising SID used to advertise the periodic advertising
All other values	Reserved for future use

Advertiser_Address_Type:**Size: 1 octet**

Value	Parameter Description
0x00	Public Device Address or Public Identity Address
0x01	Random Device Address or Random (static) Identity Address
0x02	Public Identity Address (corresponds to Resolved Private Address)
0x03	Random (static) Identity Address (corresponds to Resolved Private Address)
All other values	Reserved for future use

**Advertiser_Address:****Size: 6 octets**

Value	Parameter Description
0xFFFFFFFFXXXX	Public Device Address, Random Device Address, Public Identity Address, or Random (static) Identity Address of the advertiser

Advertiser_PHY:**Size: 1 octet**

Value	Parameter Description
0x01	Advertiser PHY is LE 1M
0x02	Advertiser PHY is LE 2M
0x03	Advertiser PHY is LE Coded
All other values	Reserved for future use

Periodic_Advertising_Interval:**Size: 2 octets**

Value	Parameter Description
N = 0xXXXX	Periodic advertising interval Range: 0x0006 to 0xFFFF Time = N * 1.25 ms Time Range: 7.5 ms to 81.91875 s

Advertiser_Clock_Accuracy:**Size: 1 octet**

Value	Parameter Description
0x00	500 ppm
0x01	250 ppm
0x02	150 ppm
0x03	100 ppm
0x04	75 ppm
0x05	50 ppm
0x06	30 ppm
0x07	20 ppm
All other values	Reserved for future use



7.7.65.25 LE CIS Established event

Event	Event Code	Event Parameters
HCI_LE_CIS_Established	0x3E	Subevent_Code, Status, Connection_Handle, CIG_Sync_Delay, CIS_Sync_Delay, Transport_Latency_C_To_P, Transport_Latency_P_To_C, PHY_C_To_P, PHY_P_To_C, NSE, BN_C_To_P, BN_P_To_C, FT_C_To_P, FT_P_To_C, Max_PDU_C_To_P, Max_PDU_P_To_C, ISO_Interval

Description:

The HCI_LE_CIS_Established event indicates that a CIS has been established, was considered lost before being established, or—on the Central—was rejected by the Peripheral. It is generated by the Controller in the Central and Peripheral. The Connection_Handle parameter shall be set to the value provided in the HCI_LE_Create_CIS command on the Central and in the HCI_LE_CIS_Request event on the Peripheral.

The CIG_Sync_Delay parameter is the maximum time, in microseconds, for transmission of PDUs of all CISes in a CIG event (see [\[Vol 6\] Part B, Section 4.5.14.1](#)).

The CIS_Sync_Delay parameter is the maximum time, in microseconds, for transmission of PDUs of the specified CIS in a CIG event (see [\[Vol 6\] Part B, Section 4.5.14.1](#)).

The Transport_Latency_C_To_P and Transport_Latency_P_To_C parameters are the actual transport latencies, in microseconds, as described in [\[Vol 6\] Part G, Section 3.2.1](#) and [\[Vol 6\] Part G, Section 3.2.2](#).

The PHY_C_To_P parameter indicates the PHY selected for packets from the Central to Peripheral.



The PHY_P_To_C parameter indicates the PHY selected for packets from the Peripheral to Central.

The NSE, BN_C_To_P, BN_P_To_C, FT_C_To_P, FT_P_To_C, Max_PDU_C_To_P, Max_PDU_P_To_C, and ISO_Interval parameters are the corresponding parameters of the CIS (see [Vol 6] Part B, Section 4.5.13.1).

If this event is generated on the Peripheral with a non-zero status, the Controller shall delete the Connection_Handle and any associated ISO data paths.

Event parameters:

Subevent_Code: *Size: 1 octet*

Value	Parameter Description
0x19	Subevent Code for HCI_LE_CIS_Established event

Status: *Size: 1 octet*

Value	Parameter Description
0x00	The CIS is successfully established.
0x01 to 0xFF	The CIS failed to be established. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Connection_Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xFFFF	Connection handle of the CIS Range: 0x0000 to 0x0EFF

CIG_Sync_Delay: *Size: 3 octets*

Value	Parameter Description
0x0000EA to 0x7FFFFFFF	The maximum time, in microseconds, for transmission of PDUs of all CISes in a CIG event
All other values	Reserved for future use

CIS_Sync_Delay: *Size: 3 octets*

Value	Parameter Description
0x0000EA to 0x7FFFFFFF	The maximum time, in microseconds, for transmission of PDUs of the specified CIS in a CIG event
All other values	Reserved for future use

*Transport_Latency_C_To_P:**Size: 3 octets*

Value	Parameter Description
0x0000EA to 0x7FFFFFFF	The actual transport latency, in microseconds, from Central to Peripheral
All other values	Reserved for future use

*Transport_Latency_P_To_C:**Size: 3 octets*

Bit Number	Parameter Description
0x0000EA to 0x7FFFFFFF	The actual transport latency, in microseconds, from Peripheral to Central
All other values	Reserved for future use

*PHY_C_To_P:**Size: 1 octet*

Value	Parameter Description
0x01	The transmitter PHY of packets from the Central is LE 1M.
0x02	The transmitter PHY of packets from the Central is LE 2M.
0x03	The transmitter PHY of packets from the Central is LE Coded.
All other values	Reserved for future use.

*PHY_P_To_C:**Size: 1 octet*

Value	Parameter Description
0x01	The transmitter PHY of packets from the Peripheral is LE 1M.
0x02	The transmitter PHY of packets from the Peripheral is LE 2M.
0x03	The transmitter PHY of packets from the Peripheral is LE Coded.
All other values	Reserved for future use.

*NSE:**Size: 1 octet*

Value	Parameter Description
0x01 to 0x1F	Maximum number of subevents in each isochronous event
All other values	Reserved for future use

**BN_C_To_P:****Size: 1 octet**

Value	Parameter Description
0x00	No isochronous data from the Central to the Peripheral
0x01 to 0x0F	The burst number for Central to Peripheral transmission
All other values	Reserved for future use

BN_P_To_C:**Size: 1 octet**

Value	Parameter Description
0x00	No isochronous data from the Peripheral to the Central
0x01 to 0x0F	The burst number for Peripheral to Central transmission
All other values	Reserved for future use

FT_C_To_P:**Size: 1 octet**

Value	Parameter Description
0xXX	The flush timeout, in multiples of the ISO_Interval for the CIS, for each payload sent from the Central to the Peripheral. Range: 0x01 to 0xFF

FT_P_To_C:**Size: 1 octet**

Value	Parameter Description
0xXX	The flush timeout, in multiples of the ISO_Interval for the CIS, for each payload sent from the Peripheral to the Central. Range: 0x01 to 0xFF

Max_PDU_C_To_P:**Size: 2 octets**

Value	Parameter Description
0x0000 to 0x00FB	Maximum size, in octets, of the payload from Central to Peripheral
All other values	Reserved for future use

Max_PDU_P_To_C:**Size: 2 octets**

Value	Parameter Description
0x0000 to 0x00FB	Maximum size, in octets, of the payload from Peripheral to Central
All other values	Reserved for future use

*ISO_Interval:**Size: 2 octets*

Value	Parameter Description
N = 0xFFFF	The time between two consecutive CIS anchor points. Range: 0x0004 to 0x0C80 Time = N * 1.25 ms Time Range: 5 ms to 4 s.

**7.7.65.26 LE CIS Request event**

Event	Event Code	Event Parameters
HCI_LE_CIS_Request	0x3E	Subevent_Code, ACL_Connection_Handle, CIS_Connection_Handle, CIG_ID, CIS_ID

Description:

The HCI_LE_CIS_Request event indicates that a Controller has received a request to establish a CIS. If the Controller receives such a request while the HCI_LE_CIS_Request event is masked away, it shall reject it. Otherwise the Controller shall assign a connection handle for the requested CIS and send the handle in the CIS_Connection_Handle parameter of the event.

When the Host receives this event it shall respond with either an HCI_LE_Accept_CIS_Request command or an HCI_LE_Reject_CIS_Request command before the timer Connection_Accept_Timeout expires. If it does not, the Controller shall reject the request and generate an HCI_LE_CIS_Established event with the status *Connection Accept Timeout Exceeded* (0x10).

The ACL_Connection_Handle is the connection handle of the ACL connection that is associated with the requested CIS.

The CIG_ID parameter contains the identifier of the CIG that contains the requested CIS. This parameter is sent by the Central in the request to establish the CIS.

The CIS_ID parameter contains the identifier of the requested CIS. This parameter is sent by the Central in the request to establish the CIS.

Event parameters:

Subevent_Code:

Size: 1 octet

Value	Parameter Description
0x1A	Subevent Code for the HCI_LE_CIS_Request event

ACL_Connection_Handle:

Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xFFFF	Connection handle of the ACL Range: 0x0000 to 0x0EFF

**CIS_Connection_Handle:****Size: 2 octets (12 bits meaningful)**

Value	Parameter Description
0xFFFF	Connection handle of the CIS Range: 0x0000 to 0x0EFF

CIG_ID:**Size: 1 octet**

Value	Parameter Description
0x00 to 0xEF	Identifier of the CIG
All other values	Reserved for future use

CIS_ID:**Size: 1 octet**

Value	Parameter Description
0x00 to 0xEF	Identifier of the CIS
All other values	Reserved for future use



7.7.65.27 LE Create BIG Complete event

Event	Event Code	Event Parameters
HCI_LE_Create_BIG_Complete	0x3E	Subevent_Code, Status, BIG_Handle, BIG_Sync_Delay, Transport_Latency_BIG, PHY, NSE, BN, PTO, IRC, Max_PDU, ISO_Interval, Num_BIS, Connection_Handle[i]

Description:

The HCI_LE_Create_BIG_Complete event indicates that the HCI_LE_-Create_BIG command has completed.

The BIG_Handle parameter shall be the same as the one specified in the HCI_LE_Create_BIG command.

The BIG_Sync_Delay parameter is the maximum time, in microseconds, for transmission of PDUs of all BISes in a BIG event (see [\[Vol 6\] Part B, Section 4.4.6.4](#)).

The Transport_Latency_BIG parameter is the actual transport latency, in microseconds, as described in [\[Vol 6\] Part G, Section 3.2.1](#) and [\[Vol 6\] Part G, Section 3.2.2](#).

The Num_BIS parameter is the total number of BISes in the BIG. This parameter shall be the same as the parameter that is provided by the Host in the HCI_LE_Create_BIG command.

The PHY parameter is the PHY used to create the BIG.

The NSE, BN, PTO, IRC, Max_PDU, and ISO_Interval parameters are the corresponding parameters of the BIS (see [\[Vol 6\] Part B, Section 4.4.6.3](#)).

The Connection_Handle arrayed parameter contains the connection handles of all the BIS in the BIG.

**Event parameters:***Subevent_Code:**Size: 1 octet*

Value	Parameter Description
0x1B	Subevent Code for the HCI_LE_Create_BIG_Complete event

*Status:**Size: 1 octet*

Value	Parameter Description
0x00	The BIG was successfully created.
0x01 to 0xFF	There was an error creating the BIG. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

*BIG_Handle:**Size: 1 octet*

Value	Parameter Description
0x00 to 0xEF	The identifier of the BIG
All other values	Reserved for future use

*BIG_Sync_Delay:**Size: 3 octets*

Value	Parameter Description
0x0000EA to 0x7FFFFFFF	The maximum time in microseconds for transmission of PDUs of all BISes in a BIG event
All other values	Reserved for future use.

*Transport_Latency_BIG:**Size: 3 octets*

Value	Parameter Description
0x0000EA to 0x7FFFFFFF	The actual transport latency, in microseconds
All other values	Reserved for future use

*PHY:**Size: 1 octet*

Value	Parameter Description
0x01	The PHY used to create the BIG is LE 1M.
0x02	The PHY used to create the BIG is LE 2M.
0x03	The PHY used to create the BIG is LE Coded.
All other values	Reserved for future use.

Host Controller Interface Functional Specification**NSE:****Size: 1 octet**

Value	Parameter Description
0x01 to 0x1F	The number of subevents in each BIS event in the BIG
All other values	Reserved for future use

BN:**Size: 1 octet**

Value	Parameter Description
0x01 to 0x07	The number of new payloads in each BIS event
All other values	Reserved for future use

PTO:**Size: 1 octet**

Value	Parameter Description
0x00 to 0x0F	Offset used for pre-transmissions
All other values	Reserved for future use

IRC:**Size: 1 octet**

Value	Parameter Description
0x01 to 0x0F	The number of times a payload is transmitted in a BIS event
All other values	Reserved for future use

Max_PDU:**Size: 2 octets**

Value	Parameter Description
0x0001 to 0x00FB	Maximum size, in octets, of the payload
All other values	Reserved for future use

ISO_Interval:**Size: 2 octets**

Value	Parameter Description
N = 0xXXXX	The time between two consecutive BIG anchor points. Range: 0x0004 to 0x0C80 Time = N * 1.25 ms Time Range: 5 ms to 4 s.

**Num_BIS:****Size: 1 octet**

Value	Parameter Description
0x01 to 0x1F	Total number of BISes in the BIG
All other values	Reserved for future use

Connection_Handle[i]:**Size: 2 octets (12 bits meaningful) * Num_BIS**

Value	Parameter Description
0xFFFF	Connection handle of a BIS Range: 0x0000 to 0x0EFF

**7.7.65.28 LE Terminate BIG Complete event**

Event	Event Code	Event Parameters
HCI_LE_Terminate_BIG_Complete	0x3E	Subevent_Code, BIG_Handle, Reason

Description:

The HCI_LE_Terminate_BIG_Complete event indicates that the transmission of all the BISes in the BIG are terminated.

The BIG_Handle parameter is used to identify the BIG that is terminated. This parameter is provided by the Host in the HCI_LE_Terminate_BIG command.

If the BIG is terminated by the local Host, the Reason parameter shall be set to the error code *Connection Terminated By Local Host* (0x16).

Event parameters:*Subevent_Code:**Size: 1 octet*

Value	Parameter Description
0x1C	Subevent Code for the HCI_LE_Terminate_BIG_Complete event

*BIG_Handle:**Size: 1 octet*

Value	Parameter Description
0x00 to 0xEF	The identifier of the BIG
All other values	Reserved for future use

*Reason:**Size: 1 octet*

Value	Parameter Description
0xXX	Reason for termination. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.



7.7.65.29 LE BIG Sync Established event

Event	Event Code	Event Parameters
HCI_LE_BIG_Sync_Established	0x3E	Subevent_Code, Status, BIG_Handle, Transport_Latency_BIG, NSE, BN, PTO, IRC, Max_PDU, ISO_Interval, Num_BIS, Connection_Handle[i]

Description:

The HCI_LE_BIG_Sync_Established event indicates that the HCI_LE_BIG_-Create_Sync command has completed.

The BIG_Handle parameter is used to identify the BIG. This parameter is provided by the Host in the HCI_LE_BIG_Create_Sync command.

The Transport_Latency_BIG parameter is the actual transport latency, in microseconds, as described in [\[Vol 6\] Part G, Section 3.2.1](#) and [\[Vol 6\] Part G, Section 3.2.2](#).

The NSE, BN, PTO, IRC, Max_PDU, and ISO_Interval parameters are the corresponding parameters of the BIS (see [\[Vol 6\] Part B, Section 4.4.6.3](#)).

The Num_BIS parameter indicates the number of BISes in the synchronized BIG specified by the HCI_LE_BIG_Create_Sync command. This parameter shall be the same as the parameter that is provided by the Host in the HCI_LE_BIG_Create_Sync command.

The Connection_Handle arrayed parameter is the list of connection handle(s) of the BIS(es) that are requested in the HCI_LE_BIG_Create_Sync command. The order of the connection handle(s) shall correspond to the order of the BIS(s) that are requested in the BIS arrayed parameter field of the HCI_LE_BIG_Create_Sync command.

**Event parameters:***Subevent_Code:**Size: 1 octet*

Value	Parameter Description
0x1D	Subevent Code for the HCI_LE_BIG_Sync_Established event

*Status:**Size: 1 octet*

Value	Parameter Description
0x00	Synchronization to the BIG is completed.
0x01 to 0xFF	Synchronization to the BIG failed to complete. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

*BIG_Handle:**Size: 1 octet*

Value	Parameter Description
0x00 to 0xEF	The identifier of the BIG
All other values	Reserved for future use

*Transport_Latency_BIG:**Size: 3 octets*

Value	Parameter Description
0x0000EA to 0x7FFFFFFF	The actual transport latency, in microseconds
All other values	Reserved for future use

*NSE:**Size: 1 octet*

Value	Parameter Description
0x01 to 0x1F	The number of subevents in each BIS event in the BIG
All other values	Reserved for future use

*BN:**Size: 1 octet*

Value	Parameter Description
0x01 to 0x07	The number of new payloads in each BIS event
All other values	Reserved for future use

Host Controller Interface Functional Specification**PTO:***Size: 1 octet*

Value	Parameter Description
0x00 to 0x0F	Offset used for pre-transmissions
All other values	Reserved for future use

IRC:*Size: 1 octet*

Value	Parameter Description
0x01 to 0x0F	The number of times a payload is transmitted in a BIS event
All other values	Reserved for future use

Max_PDU:*Size: 2 octets*

Value	Parameter Description
0x0001 to 0x00FB	Maximum size, in octets, of the payload
All other values	Reserved for future use

ISO_Interval:*Size: 2 octets*

Value	Parameter Description
N = 0xXXXX	The time between two consecutive BIG anchor points. Range: 0x0004 to 0x0C80 Time = N * 1.25 ms Time Range: 5 ms to 4 s.

Num_BIS:*Size: 1 octet*

Value	Parameter Description
0x01 to 0x1F	Total number of BISes in the BIG
All other values	Reserved for future use

Connection_Handle[i]:*Size: Num_BIS * 2 octets (12 bits meaningful)*

Value	Parameter Description
0xXXXX	Connection handle of a BIS in the BIG Range: 0x0000 to 0x0EFF



7.7.65.30 LE BIG Sync Lost event

Event	Event Code	Event Parameters
HCI_LE_BIG_Sync_Lost	0x3E	Subevent_Code, BIG_Handle, Reason

Description:

The HCI_LE_BIG_Sync_Lost event indicates that the Controller has not received any PDUs on a BIG within the timeout period BIG_Sync_Timeout or the BIG has been terminated by the remote device.

The BIG_Handle parameter is used to identify the BIG. This parameter is provided by the Host in the HCI_LE_BIG_Create_Sync command.

The Reason parameter is used to indicate the reason why the synchronization was lost or terminated. If synchronization was terminated due to the Broadcaster terminating the BIG, the Reason parameter shall be set to the error code *Remote User Terminated Connection* (0x13). If synchronization was terminated due to a timeout, the Reason parameter shall be set to the error code *Connection Timeout* (0x08). If synchronization was terminated due to a MIC failure, the Reason parameter shall be set to the error code *Connection Terminated due to MIC Failure* (0x3D).

When the HCI_LE_BIG_Sync_Lost event occurs, the Controller shall remove the connection handle(s) and data paths of all BIS(s) in the BIG with which the Controller was synchronized.

Event parameters:

Subevent_Code: *Size: 1 octet*

Value	Parameter Description
0x1E	Subevent Code for the HCI_LE_BIG_Sync_Lost event

BIG_Handle: *Size: 1 octet*

Value	Parameter Description
0x00 to 0xEF	The identifier of a BIG
All other values	Reserved for future use

Reason: *Size: 1 octet*

Value	Parameter Description
0xXX	The synchronization to BIG is terminated. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

**7.7.65.31 LE Request Peer SCA Complete event**

Event	Event Code	Event Parameters
HCI_LE_Request_Peer_SCA_Complete	0x3E	Subevent_Code, Status, Connection_Handle, Peer_Clock_Accuracy

Description:

The HCI_LE_Request_Peer_SCA_Complete event indicates that the HCI_LE_Request_Peer_SCA command has been completed.

The Peer_Clock_Accuracy parameter contains the sleep clock accuracy of the peer.

The Connection_Handle is the connection handle of the ACL connection in which the HCI_LE_Request_Peer_SCA command is issued.

Event parameters:

Subevent_Code: *Size: 1 octet*

Value	Parameter Description
0x1F	Subevent Code for the HCI_LE_Request_Peer_SCA_Complete event

Status: *Size: 1 octet*

Value	Parameter Description
0x00	The Peer_Clock_Accuracy parameter is successfully received.
0x01 to 0xFF	The reception of Peer_Clock_Accuracy parameter failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Connection_Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xFFFF	Connection handle of the ACL Range: 0x0000 to 0x0EFF

Peer_Clock_Accuracy: *Size: 1 octet*

Value	Parameter Description
0x00	251 ppm to 500 ppm
0x01	151 ppm to 250 ppm
0x02	101 ppm to 150 ppm



Value	Parameter Description
0x03	76 ppm to 100 ppm
0x04	51 ppm to 75 ppm
0x05	31 ppm to 50 ppm
0x06	21 ppm to 30 ppm
0x07	0 ppm to 20 ppm
All other values	Reserved for future use



7.7.65.32 LE Path Loss Threshold event

Event	Event Code	Event Parameters
HCI_LE_Path_Loss_Threshold	0x3E	Subevent_Code, Connection_Handle, Current_Path_Loss, Zone_Entered

Description:

The HCI_LE_Path_Loss_Threshold event is used to report a path loss threshold crossing (see [Vol 6] Part B, Section 4.5.16) on the ACL connection identified by the Connection_Handle parameter.

The Current_Path_Loss parameter indicates the current path loss value as calculated by the Controller.

The Zone_Entered parameter indicates which zone was entered. If Current_Path_Loss is set to 0xFF then Zone_Entered shall be ignored.

Event parameters:

Subevent_Code: *Size: 1 octet*

Value	Parameter Description
0x20	Subevent code for the HCI_LE_Path_Loss_Threshold event

Connection_Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

Current_Path_Loss: *Size: 1 octet*

Value	Parameter Description
0xFF	Current path loss (always zero or positive) Units: dB
0xFF	Unavailable

Zone_Entered: *Size: 1 octet*

Value	Parameter Description
0x00	Entered low zone
0x01	Entered middle zone



Value	Parameter Description
0x02	Entered high zone
All other values	Reserved for future use

**7.7.65.33 LE Transmit Power Reporting event**

Event	Event Code	Event Parameters
HCI_LE_Transmit_Power_Reporting	0x3E	Subevent_Code, Status, Connection_Handle, Reason, PHY, TX_Power_Level, TX_Power_Level_Flag, Delta

Description:

The HCI_LE_Transmit_Power_Reporting event is used to report the transmit power level on the ACL connection identified by the Connection_Handle parameter.

The Reason parameter indicates why the event was sent and the device whose transmit power level is being reported.

Whenever the transmit power changes and local reporting has been enabled by the HCI_LE_Set_Transmit_Power_Reporting_Enable command, the Controller shall generate this event with Reason set to 0x00. In this case, the PHY, TX_Power_Level, TX_Power_Level_Flag and Delta parameters shall refer to the local device and the Status parameter shall be ignored.

Whenever the Controller becomes aware that the peer's transmitter power has changed other than through an HCI_LE_Read_Remote_Transmit_Power_Level command and remote reporting has been enabled by the HCI_LE_Set_Transmit_Power_Reporting_Enable command, the Controller shall generate this event with Reason set to 0x01. In this case, the PHY, TX_Power_Level, TX_Power_Level_Flag and Delta parameters shall refer to the remote device and the Status parameter shall be ignored.

When the Reason is set to 0x02, this event indicates completion of an HCI_LE_Read_Remote_Transmit_Power_Level command. In this case, the PHY, TX_Power_Level, TX_Power_Level_Flag and Delta parameters shall refer to the remote device.

The PHY parameter shall indicate the PHY involved (which might not be the current transmit PHY for the relevant device).

The TX_Power_Level parameter shall indicate the transmit power level for the PHY.



The TX_Power_Level_Flag parameter shall indicate whether the transmit power level that is being reported has reached its minimum and/or maximum level. TX_Power_Level_Flag shall be ignored if the TX_Power_Level parameter is set to 0x7E or 0x7F.

The Delta parameter shall be set to the change in power level for the transmitter being reported, whenever it changes its transmit power level. When this event is generated with Reason set to 0x02, Delta shall be set to zero. Delta shall be ignored if the TX_Power_Level parameter is set to 0x7E.

Event parameters:

Subevent_Code: Size: 1 octet

Value	Parameter Description
0x21	Subevent code for the HCI_LE_Transmit_Power_Reporting event

Status: Size: 1 octet

Value	Parameter Description
0x00	HCI_LE_Read_Remote_Transmit_Power_Level command succeeded.
0x01 to 0xFF	HCI_LE_Read_Remote_Transmit_Power_Level command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Connection_Handle: Size: 2 octets (12 bits meaningful)

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

Reason: Size: 1 octet

Value	Parameter Description
0x00	Local transmit power changed
0x01	Remote transmit power changed
0x02	HCI_LE_Read_Remote_Transmit_Power_Level command completed
All other values	Reserved for future use

PHY: Size: 1 octet

Value	Parameter Description
0x01	LE 1M PHY
0x02	LE 2M PHY



Value	Parameter Description
0x03	LE Coded PHY with S=8 data coding
0x04	LE Coded PHY with S=2 data coding
All other values	Reserved for future use

TX_Power_Level:***Size: 1 octet***

Value	Parameter Description
0xXX	Transmit power level Range: -127 to 20 Units: dBm
0x7E	Remote device is not managing power levels on this PHY.
0x7F	Transmit power level is not available

TX_Power_Level_Flag:***Size: 1 octet***

Bit Number	Meaning
0	Transmit power level is at minimum level
1	Transmit power level is at maximum level
All other bits	Reserved for future use

Delta:***Size: 1 octet***

Value	Parameter Description
0xXX	Change in transmit power level (positive indicates increased power, negative indicates decreased power, zero indicates unchanged) Units: dB
0x7F	Change is not available or is out of range.

**7.7.65.34 LE BIGInfo Advertising Report event**

Event	Event Code	Event Parameters
HCI_LE_BIGInfo_Advertising_Report	0x3E	Subevent_Code, Sync_Handle Num_BIS, NSE, ISO_Interval, BN, PTO, IRC, Max_PDU, SDU_Interval, Max_SDU, PHY, Framing, Encryption

Description:

The HCI_LE_BIGInfo_Advertising_Report event indicates that the Controller has received an Advertising PDU that contained a BIGInfo field. If the Controller also generates an HCI_LE_Periodic_Advertising_Report event, the HCI_LE_BIGInfo_Advertising_Report event shall immediately follow that event.

An HCI_LE_BIGInfo_Advertising_Report event shall be generated even if the Controller is already synchronized to the BIG.

The Sync_Handle parameter shall identify the periodic advertising train containing the BIGInfo field and shall be the same as the corresponding field in the HCI_LE_Periodic_Advertising_Report event if one is generated.

The Num_BIS, NSE, ISO_Interval, BN, PTO, IRC, Max_PDU, SDU_Interval, Max_SDU, PHY, and Framing parameters correspond to the associated fields in the BIGInfo field of the Advertising PDU.

If the BIGInfo field indicates that the corresponding BIG is encrypted, the Encryption parameter shall be set to 0x01. Otherwise, the Encryption parameter shall be set to 0x00.

**Event parameters:***Subevent_Code:**Size: 1 octet*

Value	Parameter Description
0x22	Subevent code for the HCI_LE_BIGInfo_Advertising_Report event

*Sync_Handle:**Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xFFFF	Sync_Handle identifying the periodic advertising train. Range: 0x0000 to 0x0EFF

*Num_BIS:**Size: 1 octet*

Value	Parameter Description
0x01 to 0x1F	Value of the Num_BIS subfield of the BIGInfo field
All other values	Reserved for future use

*NSE:**Size: 1 octet*

Value	Parameter Description
0x01 to 0x1F	Value of the NSE subfield of the BIGInfo field
All other values	Reserved for future use

*ISO_Interval:**Size: 2 octets*

Value	Parameter Description
0xFFFF	Value of the ISO_Interval subfield of the BIGInfo field

*BN:**Size: 1 octet*

Value	Parameter Description
0x01 to 0x07	Value of the BN subfield of the BIGInfo field
All other values	Reserved for future use

*PTO:**Size: 1 octet*

Value	Parameter Description
0x00 to 0x0F	Value of the PTO subfield of the BIGInfo field
All other values	Reserved for future use

**IRC:****Size: 1 octet**

Value	Parameter Description
0x01 to 0x0F	Value of the IRC subfield of the BIGInfo field
All other values	Reserved for future use

Max_PDU:**Size: 2 octets**

Value	Parameter Description
0x0001 to 0x00FB	Value of the Max_PDU subfield of the BIGInfo
All other values	Reserved for future use

SDU_Interval:**Size: 3 octets**

Value	Parameter Description
0x0000FF to 0x0FFFFFF	Value of the SDU_Interval subfield of the BIGInfo field
All other values	Reserved for future use

Max_SDU:**Size: 2 octets**

Value	Parameter Description
0x0001 to 0x0FFF	Value of the Max_SDU subfield of the BIGInfo field in the Advertising PDU
All other values	Reserved for future use

PHY:**Size: 1 octet**

Value	Parameter Description
0x01	The BIG is transmitted on the LE 1M PHY
0x02	The BIG is transmitted on the LE 2M PHY
0x03	The BIG is transmitted on the LE Coded PHY
All other values	Reserved for future use

Framing:**Size: 1 octet**

Value	Parameter Description
0x00	Unframed
0x01	Framed
All other values	Reserved for future use

Host Controller Interface Functional Specification*Encryption:**Size: 1 octet*

Value	Parameter Description
0x00	BIG carries unencrypted data
0x01	BIG carries encrypted data
All other values	Reserved for future use

**7.7.65.35 LE Subrate Change event**

Event	Event Code	Event Parameters
HCI_LE_Subrate_Change	0x3E	Subevent_Code, Status, Connection_Handle, Subrate_Factor, Peripheral_Latency, Continuation_Number, Supervision_Timeout

Description:

The HCI_LE_Subrate_Change event is used to indicate that a Connection Subrate Update procedure has completed and some parameters of the specified connection have changed.

This event shall be issued if the HCI_LE_Subrate_Request command was issued by the Host or the parameters are updated successfully following a request from the peer device. If no parameters are updated following a request from the peer device or the parameters were changed using the Connection Update procedure, then this event shall not be issued.

Event parameters:

Subevent_Code: *Size: 1 octet*

Value	Parameter Description
0x23	Subevent Code for the HCI_LE_Subrate_Change event

Status: *Size: 1 octet*

Value	Parameter Description
0x00	The HCI_LE_Subrate_Request command succeeded or this event was generated following a request from the peer device.
0x01 to 0xFF	The HCI_LE_Subrate_Request command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

Connection_Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xFFFF	Connection handle Range: 0x0000 to 0x0EFF

**Subrate_Factor:****Size: 2 octets**

Value	Parameter Description
0xFFFF	New subrate factor applied to the specified underlying connection interval Range 0x0001 to 0x01F4
All other values	Reserved for future use

Peripheral_Latency:**Size: 2 octets**

Value	Parameter Description
0xFFFF	New Peripheral latency for the connection in number of subrated connection events Range: 0x0000 to 0x01F3
All other values	Reserved for future use

Continuation_Number:**Size: 2 octets**

Value	Parameter Description
0xFFFF	Number of underlying connection events to remain active after a packet containing a Link Layer PDU with a non-zero Length field is sent or received Range: 0x0000 to 0x01F3
All other values	Reserved for future use

Supervision_Timeout:**Size: 2 octets**

Value	Parameter Description
N = 0xFFFF	New supervision timeout for this connection. Range: 0x000A to 0x0C80 Time = N × 10 ms Time Range: 100 ms to 32 s



7.7.66 Triggered Clock Capture event

Event	Event Code	Event Parameters
HCI_Triggered_Clock_Capture	0x4E	Connection_Handle, Which_Clock, Clock, Slot_Offset

Description:

The HCI_Triggered_Clock_Capture event is sent to indicate that a triggering event has occurred at the specified clock and offset value. The Which_Clock parameter indicates whether the clock is local or a piconet clock. The Connection_Handle parameter is used when the clock is a piconet clock to indicate which piconet's clock was reported.

The Clock parameter indicates the value of the selected clock at the instant of the triggering event, with bits 1 and 0 set to 0b00.

The Slot_Offset parameter indicates the number of microseconds (from 0 to 1249) from the instant at which the selected clock took the value Clock until the triggering event.

Note: What constitutes a triggering event is defined by the Controller implementation. For example, it could be an interrupt signal received by the Controller hardware.

Event parameters:

Connection_Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xFFFF	Connection_Handle Range: 0x0000 to 0x0EFF

Which_Clock: *Size: 1 octet*

Value	Parameter Description
0xFF	0x00 = Local Clock (Connection_Handle does not have to be valid) 0x01 = Piconet Clock (Connection_Handle shall be valid) 0x02 to 0xFF = Reserved for future use

Clock: *Size: 4 octets (28 bits meaningful)*

Value	Parameter Description
0xFFFFFFFF	Bluetooth clock of the device requested with bits 1 and 0 set to 0b00.



Slot_Offset:

Size: 2 octets

Value	Parameter Description
0xFFFF	Number of microseconds from the selected clock attaining the value Clock until the triggering event. Range: 0 to 1249.



7.7.67 Synchronization Train Complete event

Event	Event Code	Event Parameters
HCI_Synchronization_Train_Complete	0x4F	Status

Description:

The HCI_Synchronization_Train_Complete event indicates that the HCI_Start_Synchronization_Train command has completed.

Event parameters:*Status:**Size: 1 octet*

Value	Parameter Description
0x00	HCI_Start_Synchronization_Train command completed successfully.
0x01 to 0xFF	HCI_Start_Synchronization_Train command failed. See [Vol 1] Part F, Controller Error Codes , for error codes and descriptions.



7.7.68 Synchronization Train Received event

Event	Event Code	Event Parameters
HCI_Synchronization_Train_Received	0x50	Status, BD_ADDR, Clock_Offset, AFH_Channel_Map, LT_ADDR, Next_Broadcast_Instant, Connectionless_Peripheral_- Broadcast_Interval, Service_Data

Description:

The HCI_Synchronization_Train_Received event provides information received from a synchronization train packet transmitted by a Connectionless Peripheral Broadcast transmitter with the given BD_ADDR. If synchronization was successful, it provides the clock offset, AFH channel map, LT_ADDR, next broadcast instant, broadcast interval, and service data as received from the synchronization train payload. If the command returns a status of 0x01 to 0xFF, then all other parameters are undefined and shall be ignored.

A packet with the Connectionless Peripheral Broadcast LT_ADDR field in the payload set to zero shall be ignored for the purposes of this event.

Event parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Receive_Synchronization_Train command completed successfully.
0x01 to 0xFF	HCI_Receive_Synchronization_Train command failed. See [Vol 1 Part F] , for error codes and descriptions.

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0XXXXXXXXXXXXX	BD_ADDR of the Connectionless Peripheral Broadcast transmitter.

Clock_Offset:

Size: 4 octets (28 bits meaningful)

Value	Parameter Description
0XXXXXXXX	(CLKNreceiver - CLKNtransmitter) $\text{mod } 2^{28}$

**AFH_Channel_Map:****Size: 10 octets (79 bits meaningful)**

Value	Parameter Description
0XXXXXXXXX XXXXXXXXXX XX	<p>This parameter contains 80 1-bit fields.</p> <p>The n^{th} such field (in the range 0 to 78) contains the value for channel n:</p> <p>0: channel n is unused 1: channel n is used</p> <p>The most significant bit (bit 79) is reserved for future use</p>

LT_ADDR:**Size: 1 octet**

Value	Parameter Description
0x01 to 0x07	LT_ADDR of Connectionless Peripheral Broadcast channel.
All other values	Reserved for future use

Next_Broadcast_Instant:**Size: 4 octets (28 bits meaningful)**

Value	Parameter Description
0XXXXXXXXX	CLK of a future broadcast on this channel

Connectionless_Peripheral_Broadcast_Interval:**Size: 2 octets**

Value	Parameter Description
0XXXXX	<p>Interval between Connectionless Peripheral Broadcast instants in slots.</p> <p>Range: 0x0002 to 0xFFFE; only even values are valid</p>

Service_Data:**Size: 1 octet**

Value	Parameter Description
0xXX	Value from octet 27 of the Synchronization Train packet; see [Vol 2] Part B, Table 8.11 .



7.7.69 Connectionless Peripheral Broadcast Receive event

Event	Event Code	Event Parameters
HCI_Connectionless_Peripheral_Broadcast_Receive	0x51	BD_ADDR, LT_ADDR, Clock, Offset, RX_Status, Fragment, Data_Length, Data

Description:

The HCI_Connectionless_Peripheral_Broadcast_Receive event shall be sent by the BR/EDR Controller every Connectionless Peripheral Broadcast Instant on which the BR/EDR Controller is scheduled to receive a Connectionless Peripheral Broadcast packet. If the packet is not received successfully, the event returns a RX_Status of 0x01. Otherwise, the event returns the payload Data along with the Piconet Clock and the offset from the local CLKN when the packet was received.

The BR/EDR Controller shall send multiple HCI_Connectionless_Peripheral_Broadcast_Receive events if the length of the received data exceeds the capacity of a single HCI_Connectionless_Peripheral_Broadcast_Receive event. The fragments shall be marked as starting, continuation, or ending to allow the Host to reassemble the received packet. Only a single event shall be generated for a Connectionless Peripheral Broadcast instant on which a Connectionless Peripheral Broadcast packet was scheduled for reception but the BR/EDR Controller failed to successfully receive it.

Event parameters:

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0XXXXXXXXXXXXX	BD_ADDR of the broadcasting (transmitter) device

LT_ADDR:

Size: 1 octet

Value	Parameter Description
0x01 to 0x07	LT_ADDR of the Connectionless Peripheral Broadcast
All other values	Reserved for future use

**Clock:****Size: 4 octets (28 bits meaningful)**

Value	Parameter Description
0xFFFFFFFF	CLK when Connectionless Peripheral Broadcast data was received

Offset:**Size: 4 octets (28 bits meaningful)**

Value	Parameter Description
0xFFFFFFFF	(CLKNreceiver - CLKNtransmitter) $\text{mod } 2^{28}$

RX_Status:**Size: 1 octet**

Value	Parameter Description
0x00	Packet received successfully
0x01	Packet not received successfully (Fragment, Data_Length, and Data fields invalid)
All other values	Reserved for future use

Fragment:**Size: 1 octet**

Value	Parameter Description
0x00	Continuation fragment
0x01	Starting fragment
0x02	Ending fragment
0x03	No fragmentation (single fragment)
All other values	Reserved for future use

Data_Length:**Size: 1 octet**

Value	Parameter Description
0xFF	Length of Data field

Data:**Size: Data_Length octets**

Value	Parameter Description
Variable	Data received from a Connectionless Peripheral Broadcast packet.



7.7.70 Connectionless Peripheral Broadcast Timeout event

Event	Event Code	Event Parameters
HCI_Connectionless_Peripheral_Broadcast_Timeout	0x52	BD_ADDR, LT_ADDR

Description:

On the Connectionless Peripheral Broadcast Receiver, the HCI_Connectionless_Peripheral_Broadcast_Timeout event indicates to the Host that the BR/EDR Controller has lost synchronization with the Connectionless Peripheral Broadcast because no Connectionless Peripheral Broadcast packets have been received for the timeout interval, *CPB_supervisionTO*, specified in the HCI_Set_Connectionless_Peripheral_Broadcast_Receive command.

On the Connectionless Peripheral Broadcast Transmitter, the HCI_Connectionless_Peripheral_Broadcast_Timeout event indicates to the Host that the BR/EDR Controller has been unable to transmit a Connectionless Peripheral Broadcast packet for the timeout interval, *CPB_supervisionTO*, specified in the HCI_Set_Connectionless_Peripheral_Broadcast command.

Event parameters:

BD_ADDR: *Size: 6 octets*

Value	Parameter Description
0XXXXXXXXXXXXX	BD_ADDR of the broadcasting (transmitter) device

LT_ADDR: *Size: 1 octet*

Value	Parameter Description
0x01 to 0x07	LT_ADDR of the Connectionless Peripheral Broadcast
All other values	Reserved for future use



7.7.71 Truncated Page Complete event

Event	Event Code	Event Parameters
HCI_Truncated_Page_Complete	0x53	Status, BD_ADDR

Description:

The HCI_Truncated_Page_Complete event indicates to the Host that an HCI_Truncated_Page command completed. Truncated Paging is considered to be successful when a Peripheral page response ID packet has been received by the local BR/EDR Controller. See [\[Vol 2\] Part B, Section 8.3.3](#) for more information.

An HCI_Truncated_Page_Complete event shall always be sent for each HCI_Truncated_Page command. If the Host issues an HCI_Truncated_Page_Cancel command before the Controller returns the HCI_Truncated_Page_Complete event, then the HCI_Truncated_Page_Complete event shall be sent after the HCI_Command_Complete event for the HCI_Truncated_Page_Cancel command. If the cancellation was successful, the HCI_Truncated_Page_Complete event shall be generated with the error code *Unknown Connection Identifier* (0x02).

Event parameters:

Status:

Size: 1 octet

Value	Parameter Description
0x00	HCI_Truncated_Page command completed successfully.
0x01 to 0xFF	HCI_Truncated_Page command failed. See [Vol 1] Part F, Controller Error Codes for a list of error codes and descriptions.

BD_ADDR:

Size: 6 octets

Value	Parameter Description
0xFFFFFFFFXXXX	BD_ADDR of the paged (Peripheral) device



7.7.72 Peripheral Page Response Timeout event

Event	Event Code	Event Parameters
HCI_Peripheral_Page_Response_ - Timeout	0x54	

Description:

The HCI_Peripheral_Page_Response_Timeout event indicates to the Host that a Peripheral page response timeout has occurred in the BR/EDR Controller.

Note: This event will be generated if the Peripheral BR/EDR Controller responds to a page but does not receive the Central FHS packet (see [\[Vol 2\] Part B, Section 8.3.3](#)) within *pagerespTO*.

Event parameters:

None



7.7.73 Connectionless Peripheral Broadcast Channel Map Change event

Event	Event Code	Event Parameters
HCI_Connectionless_Peripheral_Broadcast_Channel_Map_Change	0x55	Channel_Map

Description:

The HCI_Connectionless_Peripheral_Broadcast_Channel_Map_Change event is sent by the Transmitter's BR/EDR Controller to the Transmitter's Host to indicate that the Transmitter's BR/EDR Controller has moved to a new AFH channel map for the PBD logical link.

After an AFH channel map change takes effect for the PBD logical link, the Connectionless Peripheral Broadcast Transmitter BR/EDR Controller shall send this event to the Host. Upon reception of this event, the Host may restart the synchronization train to allow receivers to obtain the updated AFH channel map.

This event shall also be sent if the Host issues an HCI_Set_AFH_Host_Channel_Classification command which causes the Connectionless Peripheral Broadcast Channel Map to change.

Event parameters:

Channel_Map:

Size: 10 octets (79 bits meaningful)

Value	Parameter Description
0xFFFFFFFF XXXXXXXXXX XX	This parameter contains 80 1-bit fields. The n^{th} such field (in the range 0 to 78) contains the value for channel n : 0: channel n is unused 1: channel n is used The most significant bit (bit 79) is reserved for future use



7.7.74 Inquiry Response Notification event

Event	Event Code	Event Parameters
HCI_Inquiry_Response_Notification	0x56	LAP, RSSI

Description:

The HCI_Inquiry_Response_Notification event indicates to the Host that the BR/EDR Controller responded to an Inquiry message. The LAP parameter in the event indicates the LAP used to create the access code received. The parameter may be used by the Host to determine which access code was used in cases where the BR/EDR Controller is performing inquiry scan on multiple inquiry access codes using parallel scanning or sequential scanning. See [\[Vol 3\] Part C, Section 4.1.2.1](#) for details on sequential and parallel scanning.

The LAP parameter returned by the BR/EDR Controller shall be one of the LAPs currently enabled. LAPs are enabled via the HCI_Write_Current_IAC_-LAP command.

The RSSI parameter indicates the signal strength of the received ID packet.

Event parameters:

LAP:

Size: 3 octets

Value	Parameter Description
0xXXXXXX	The LAP from which the IAC was derived; see Assigned Numbers . Range: 0x9E8B00 to 0x9E8B3F

RSSI:

Size: 1 octet

Value	Parameter Description
0xXX	Range: -100 to 20, +127 indicates unknown RSSI Units: dBm



7.7.75 Authenticated Payload Timeout Expired event

Event	Event Code	Event Parameters
HCI_Authenticated_Payload_Timeout_Expired	0x57	Connection_Handle

Description:

The HCI_Authenticated_Payload_Timeout_Expired event is used to indicate that a packet containing a valid MIC on the Connection_Handle was not received within the *authenticatedPayloadTO* (see [\[Vol 2\] Part B, Appendix B](#) for the BR/EDR and [\[Vol 6\] Part B, Section 5.4](#) for the LE connection).

Note: A Host may choose to disconnect the link when this occurs.

Event parameters:

Connection_Handle:

Size: 2 octet (12 bits meaningful)

Value	Parameter Description
0xXXXX	Connection_Handle Range: 0x0000 to 0x0EFF



7.7.76 SAM Status Change event

Event	Event Code	Event Parameters
HCI_SAM_Status_Change	0x58	Connection_Handle, Local_SAM_Index, Local_SAM_TX_Availability, Local_SAM_RX_Availability, Remote_SAM_Index, Remote_SAM_TX_Availability, Remote_SAM_RX_Availability

Description:

The HCI_SAM_Status_Change event indicates that the Controller has changed the SAM status for the connection identified by the Connection_Handle; i.e., a new SAM slot map has been enabled or the existing one disabled.

Note: A change from one SAM slot map to another only generates one event, not two.

Event parameters:

Connection_Handle: *Size: 2 octets (12 bits meaningful)*

Value	Parameter Description
0xXXXX	Connection_Handle Range: 0x0000 to 0x0EFF

Local_SAM_Index: *Size: 1 octet*

Value	Parameter Description
0xFF	The index of the current SAM slot map used by the local device. 0xFF means SAM is disabled (i.e. all slots are available)

Local_SAM_TX_Availability: *Size: 1 octet*

Value	Parameter Description
0xFF	The proportion of slots available for the local device to transmit. 0 represents "less than 1 in 255" and 255 represents "all"; other proportions shall be linearly scaled. That is, $Local_SAM_TX_Availability = (total_available_TX_slots / local_T_{SAM}) * 255$, rounded to the integer below, where total_available_TX_slots is the total number of slots available for transmission in the current local SAM slot map and local_TSAM is T _{SAM} for the current local SAM slot map.

*Local_SAM_RX_Availability:**Size: 1 octet*

Value	Parameter Description
0xXX	The proportion of slots available for the local device to receive. 0 represents "less than 1 in 255" and 255 represents "all"; other proportions shall be linearly scaled. That is, $\text{Local_SAM_RX_Availability} = (\text{total_available_RX_slots} / \text{local_T}_{\text{SAM}}) * 255$, rounded to the integer below, where $\text{total_available_RX_slots}$ is the total number of slots available for reception in the current local SAM slot map and $\text{local_T}_{\text{SAM}}$ is T_{SAM} for the current local SAM slot map.

*Remote_SAM_Index:**Size: 1 octet*

Value	Parameter Description
0xXX	The index of the current SAM slot map used by the remote device. 0xFF means SAM is disabled (i.e. all slots are available)

*Remote_SAM_TX_Availability:**Size: 1 octet*

Value	Parameter Description
0xXX	The proportion of slots available for the remote device to transmit. 0 represents "less than 1 in 255" and 255 represents "all"; other proportions shall be linearly scaled. That is, $\text{Remote_SAM_TX_Availability} = (\text{total_available_TX_slots} / \text{remote_T}_{\text{SAM}}) * 255$, rounded to the integer below, where $\text{total_available_TX_slots}$ is the total number of slots available for transmission in the current remote SAM slot map and $\text{remote_T}_{\text{SAM}}$ is T_{SAM} for the current remote SAM slot map.

*Remote_SAM_RX_Availability:**Size: 1 octet*

Value	Parameter Description
0xXX	The proportion of slots available for the remote device to receive. 0 represents "less than 1 in 255" and 255 represents "all"; other proportions shall be linearly scaled. That is, $\text{Remote_SAM_RX_Availability} = (\text{total_available_RX_slots} / \text{remote_T}_{\text{SAM}}) * 255$, rounded to the integer below, where $\text{total_available_RX_slots}$ is the total number of slots available for reception in the current remote SAM slot map and $\text{remote_T}_{\text{SAM}}$ is T_{SAM} for the current remote SAM slot map.



7.8 LE CONTROLLER COMMANDS

The LE Controller commands provide access and control to various capabilities of the Bluetooth hardware, as well as methods for the Host to affect how the Link Layer manages the piconet and controls connections.

For the LE Controller commands, the OGF code is defined as 0x08.

7.8.1 LE Set Event Mask command

Command	OCF	Command Parameters	Return Parameters
HCI_LE_Set_Event_Mask	0x0001	LE_Event_Mask	Status

Description:

The HCI_LE_Set_Event_Mask 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 a one, then the event associated with that bit will be enabled. The event mask allows the Host to control which events will interrupt it.

The Controller shall ignore those bits which are reserved for future use or represent events which it does not support. If the Host sets any of these bits to 1, the Controller shall act as if they were set to 0.

For LE events to be generated, the LE Meta event bit in the Event_Mask shall also be set. If that bit is not set, then LE events shall not be generated, regardless of how the LE_Event_Mask is set.

Command parameters:

LE_Event_Mask:

Size: 8 octets

Bit	LE Subevent Types
0	LE Connection Complete event
1	LE Advertising Report event
2	LE Connection Update Complete event
3	LE Read Remote Features Complete event
4	LE Long Term Key Request event
5	LE Remote Connection Parameter Request event
6	LE Data Length Change event
7	LE Read Local P-256 Public Key Complete event
8	LE Generate DHKey Complete event
9	LE Enhanced Connection Complete event



Bit	LE Subevent Types
10	LE Directed Advertising Report event
11	LE PHY Update Complete event
12	LE Extended Advertising Report event
13	LE Periodic Advertising Sync Established event
14	LE Periodic Advertising Report event
15	LE Periodic Advertising Sync Lost event
16	LE Scan Timeout event
17	LE Advertising Set Terminated event
18	LE Scan Request Received event
19	LE Channel Selection Algorithm event
20	LE Connectionless IQ Report event
21	LE Connection IQ Report event
22	LE CTE Request Failed event
23	LE Periodic Advertising Sync Transfer Received event
24	LE CIS Established event
25	LE CIS Request event
26	LE Create BIG Complete event
27	LE Terminate BIG Complete event
28	LE BIG Sync Established event
29	LE BIG Sync Lost event
30	LE Request Peer SCA Complete event
31	LE Path Loss Threshold event
32	LE Transmit Power Reporting event
33	LE BIGInfo Advertising Report event
34	LE Subrate Change event
60 to 63	Reserved for future use (used for specification development purposes)

The value with all bits set to 0 indicates that no events are specified. The default is for bits 0 to 4 (the value 0x0000 0000 0000 001F) to be set.

All bits not listed in this table are reserved for future use.