Part H:1

HOST CONTROLLER INTERFACE FUNCTIONAL SPECIFICATION

This document describes the functional specification for the Host Controller Interface (HCI). The HCI provides a command interface to the baseband controller and link manager, and access to hardware status and control registers. This interface provides a uniform method of accessing the Bluetooth baseband capabilities.

Bluetooth™

CONTENTS

1 INTRODUCTION

This document describes the functional specifications for the Host Controller Interface (HCI). The HCI provides a uniform interface method of accessing the Bluetooth hardware capabilities. The next two sections provide a brief overview of the lower layers of the Bluetooth software stack and of the Bluetooth hardware. Section 2, provides an overview of the Lower HCI Device Driver Interface on the host device. Section 3, describes the flow control used between the Host and the Host Controller. Section 4, describes each of the HCI Commands in details, identifies parameters for each of the commands, and lists events associated with each command.

1.1 LOWER LAYERS OF THE BLUETOOTH SOFTWARE STACK
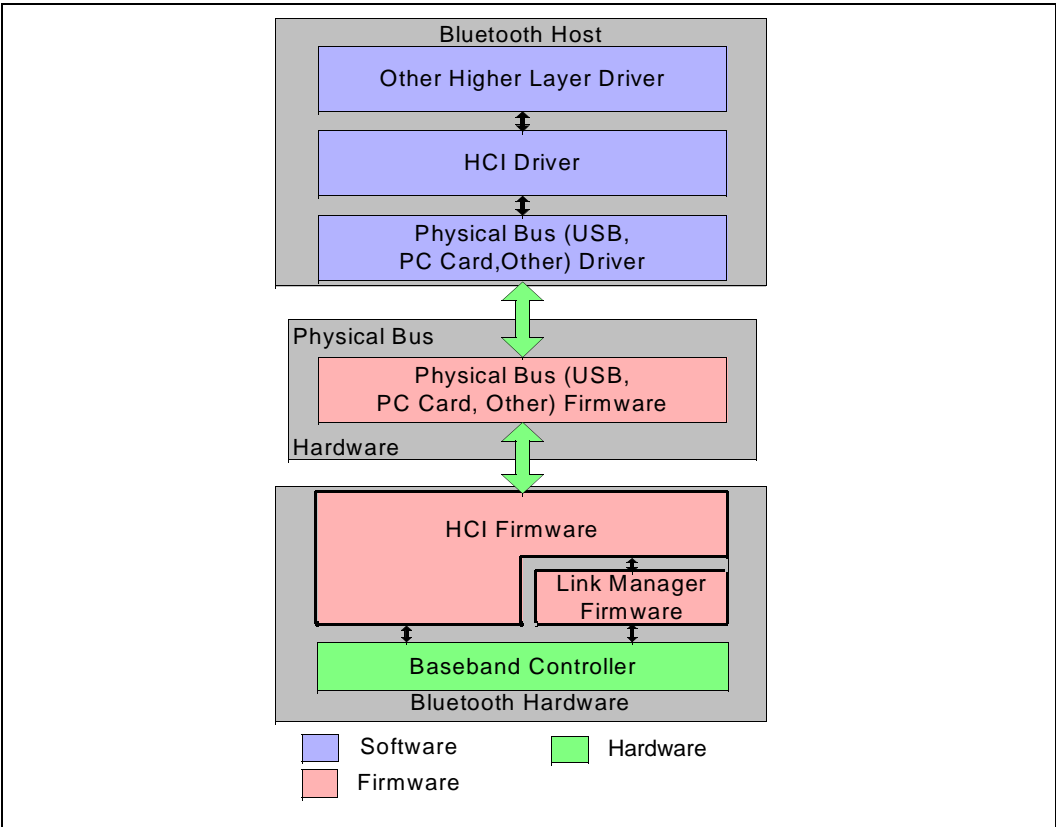
Figure 1.1: Overview of the Lower Software Layers

Figure 1.1, provides an overview of the lower software layers. The HCI firmware implements the HCI Commands for the Bluetooth hardware by accessing baseband commands link manager commands, hardware status registers, control registers, and event registers.

Several layers may exist between the HCI driver on the host system and the HCI firmware in the Bluetooth hardware. These intermediate layers, the Host Controller Transport Layer, provide the ability to transfer data without intimate knowledge of the data.
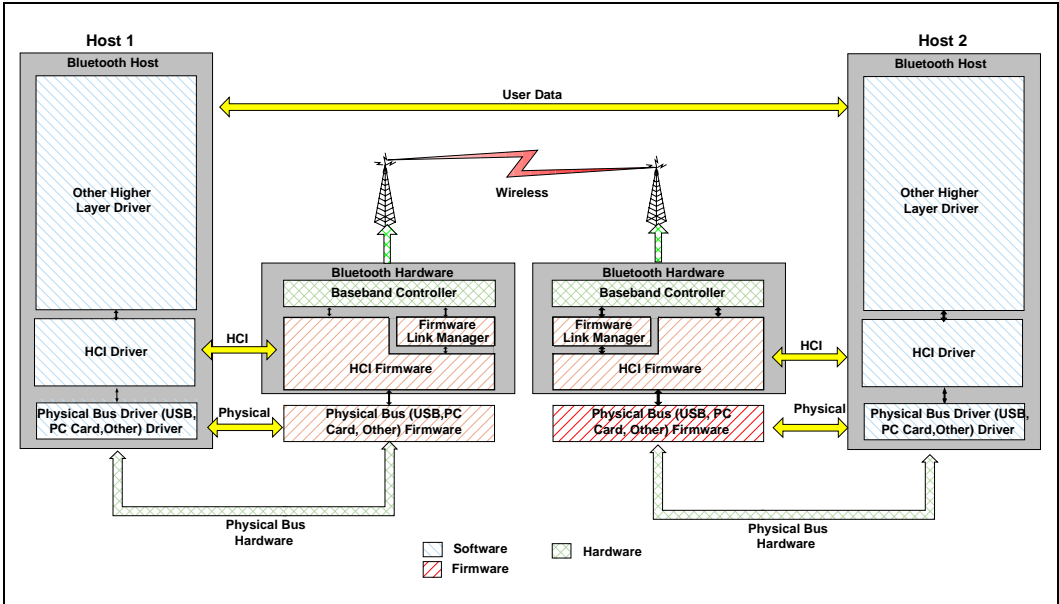
Figure 1.2: End to End Overview of Lower Software Layers to Transfer Data

Figure 1.2, illustrates the path of a data transfer from one device to another. The HCI driver on the Host exchanges data and commands with the HCI firmware on the Bluetooth hardware. The Host Control Transport Layer (i.e. physical bus) driver provides both HCI layers with the ability to exchange information with each other.

The Host will receive asynchronous notifications of HCI events independent of which Host Controller Transport Layer is used. HCI events are used for notifying the Host when something occurs. When the Host discovers that an event has occurred it will then parse the received event packet to determine which event occurred.

1.2   BLUETOOTH HARDWARE BLOCK DIAGRAM

A general overview of the Bluetooth hardware is outlined in Figure 1.3 on page 526. It consists of an analog part – the Bluetooth radio, and a digital part – the Host Controller. The Host Controller has a hardware digital signal processing part –the Link Controller (LC), a CPU core, and it interfaces to the host environment. The hardware and software parts of the Host Controller are described below.
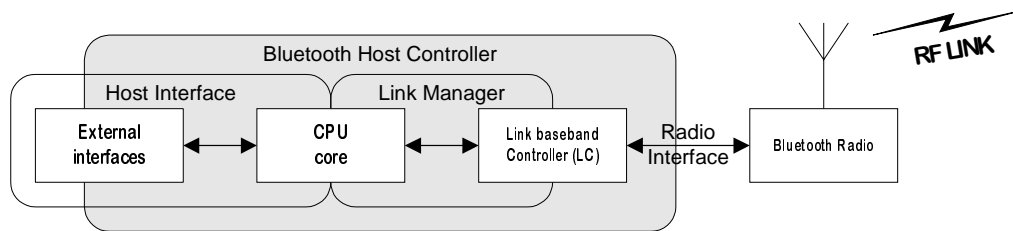
Figure 1.3: Bluetooth Hardware Architecture Overview.

1.2.1 Link Controller

The Link Controller (LC) consists of hardware and software parts that perform Bluetooth baseband processing, and physical layer protocols such as ARQ-protocol and FEC coding.

The functions performed by the Link Controller include:

- Transfer types with selected Quality-of-Service (QoS) parameters

- Asynchronous transfers with guaranteed delivery using hardware fast Automatic Repeat reQuest (fARQ). Frames can be flushed from the retransmission buffer, for use with isochronous data

- Synchronous transfers

- Audio coding. A power-efficient hardware implementation of a robust 64 Kbits/s Continuous Variable Slope Delta (CVSD) coding, as well as 64 Kbits/s log-PCM

- Encryption

1.2.2 CPU Core

The CPU core will allow the Bluetooth module to handle Inquiries and filter Page requests without involving the host device. The Host Controller can be programmed to answer certain Page messages and authenticate remote links.

The Link Manager (LM) software runs on the CPU Core. The LM discovers other remote LMs and communicates with them via the Link Manager Protocol (LMP) to perform its service provider role using the services of the underlying Link Controller (LC). For details see "Link Manager Protocol" on page 185

Bluetooth

1.3   POSSIBLE PHYSICAL BUS ARCHITECTURES

Bluetooth devices will have various physical bus interfaces that could be used to connect to the Bluetooth hardware. These buses may have different archi-tectures and different parameters. The Bluetooth Host Controller will initially support two physical bus architectures, USB, and PC Card.

1.3.1 USB HCI Architecture

The following block diagram shows the Bluetooth connection to the Host PC via the USB HCI. USB can handle several logic channels over the same single physical channel (via Endpoints). Therefore control, data, and voice channels do not require any additional physical interfaces. Note that there is no direct access to registers/memory on the Bluetooth module over USB. Instead, this is done by using the appropriate HCI Commands and by using the Host Control-ler Transport Layer interface.
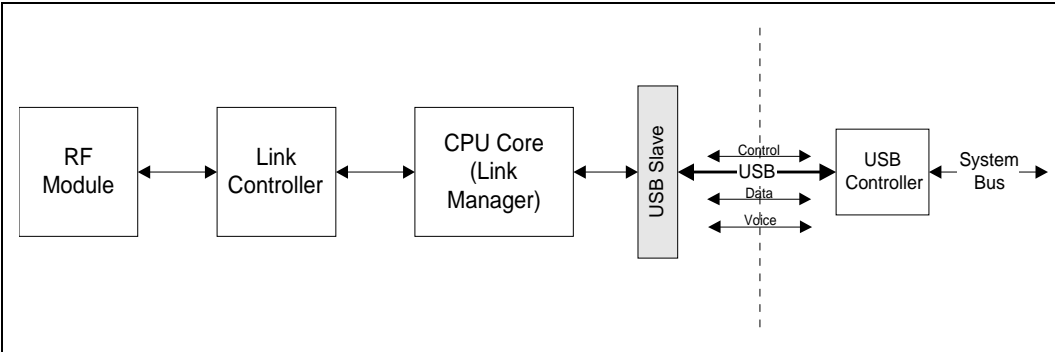
Figure 1.4: Bluetooth Block Diagram with USB HCI

1.3.2 PC Card HCI Architecture

Besides the USB interface, derivatives of the ISA bus (Compact Flash/PC Card interfaces) are an option for an integrated PC solution. Unlike USB, all traffic between the Host and the Bluetooth module will go across the PC Card bus interface. Communications between the host PC and the Bluetooth module will be primarily done directly via registers/memory. The following block diagram shows the data flow for a PC-Card HCI.
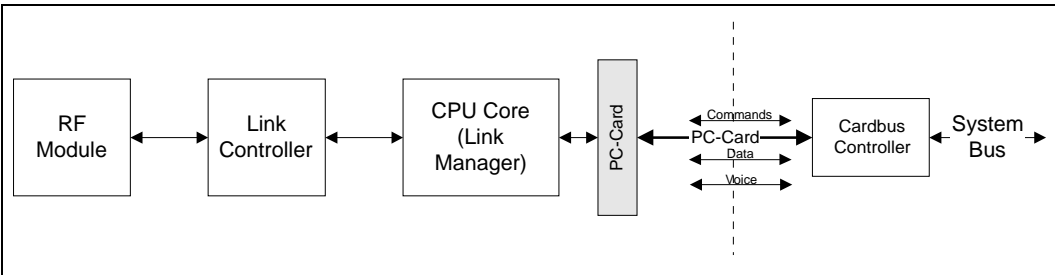
Figure 1.5: Bluetooth Block Diagram with PC-Card HCI

2 OVERVIEW OF HOST CONTROLLER TRANSPORT LAYER

The host driver stack has a transport layer between the Host Controller driver and the Host Controller. On a laptop, this transport layer might be PC Card or Universal Serial Bus (USB).

The main goal of this transport layer is transparency. The Host Controller driver (which talks to the Host Controller) should not care whether it is running over USB or a PC Card. Nor should USB or PC Card require any visibility into the data that the Host Controller driver passes to the Host Controller. This allows the interface (HCI) or the Host Controller to be upgraded without affecting the transport layer.

The Host Controller Transport Layer is described in separate documents for each physical media.

Host Controller Interface Functional Specification

Bluetooth

3 HCI FLOW CONTROL

Flow control is used in the direction from the Host to the Host Controller to avoid filling up the Host Controller data buffers with ACL data destined for a remote device (connection handle) that is not responding. It is the Host that manages the data buffers of the Host Controller.

On Initialization, the Host will issue the Read_Buffer_Size command. Two of the return parameters of this command determine the maximum size of HCI ACL and SCO Data Packets (excluding header) sent from the Host to the Host Controller. There are also two additional return parameters that specify the total number of HCI ACL and SCO Data Packets that the Host Controller can have waiting for transmission in its buffers. When there is at least one connection to another device, or when in local loopback mode, the Host Controller uses the Number Of Completed Packets event to control the flow of data from the Host. This event contains a list of connection handles and a corresponding number of HCI Data Packets that have been completed (transmitted, flushed, or looped back to the Host) since the previous time the event was returned (or since the connection was established, if the event has not been returned before for a particular connection handle). Based on the information returned in this event, and the return parameters of the Read_Buffer_Size command that specify the total number of HCI ACL and SCO Data Packets that can be stored in the Host Controller, the Host can decide for which Connection Handles the following HCI Data Packets should be sent. After every time it has sent an HCI Data Packet, the Host must assume that the free buffer space for the corresponding link type (ACL or SCO) in the Host Controller has decreased by one HCI Data Packet. When the Host receives a new Number Of Completed Packets event, the Host gets information about how much the buffer usage has decreased since the previous time the event was returned. It can then calculate the actual current buffer usage. While the Host Controller has HCI data packets in its buffer, it must keep sending the Number Of Completed Packets event to the Host at least periodically, until it finally reports that all the pending ACL Data Packets have been transmitted or flushed. The rate with which this event is sent is manufacturer specific. Note that Number Of Completed Packets events will not report on SCO connection handles if SCO Flow Control is disabled. (See Read/Write_SCO_Flow_Control_Enable on .)

Note that for each individual Connection Handle, the data must be sent to the Host Controller in HCI Data Packets in the order in which it was created in the Host. The Host Controller must also transmit data on the air that is received from the Host for a given Connection Handle in the same order as it is received from the Host. Furthermore, data that is received on the air from another device must, for the corresponding Connection Handle, be sent in HCI Data Packets to the Host in the same order as it is received. This means that the scheduling is made on a Connection Handle basis. For each individual Connection Handle, the order of the data must not be changed from the order in which the data has been created.

In certain cases, flow control may also be necessary in the direction from the Host Controller to the Host. There is therefore a command – Set_Host_Controller_To_Host_Flow_Control – to turn flow control on or off in that direction. If turned on, it works in exactly the same way as described above. On initialization, the Host uses the Host_Buffer_Size command to notify the Host Controller about the maximum size of HCI ACL and SCO Data Packets sent from the Host Controller to the Host. The command also contains two additional command parameters to notify the Host Controller about the total number of ACL and SCO Data Packets that can be stored in the data buffers of the Host. The Host then uses the Host_Number_Of_Completed_Packets command in exactly the same way as the Host Controller uses the Number Of Completed Packets event (as was previously described in this section). The Host_Number_Of_Completed_Packets command is a special command for which no command flow control is used, and which can be sent anytime there is a connection or when in local loopback mode. This makes it possible for the flow control to work in exactly the same way in both directions, and the flow of normal commands will not be disturbed.

When the Host receives a Disconnection Complete event, the Host can assume that all HCI Data Packets that have been sent to the Host Controller for the returned Connection_Handle have been flushed, and that the corresponding data buffers have been freed. The Host Controller does not have to notify the Host about this in a Number Of Completed Packets event. If flow control is also enabled in the direction from the Host Controller to the Host, the Host Controller can after it has sent a Disconnection_Complete event assume that the Host will flush its data buffers for the sent Connection_Handle when it receives the Disconnection_Complete event. The Host does not have to notify the Host Controller about this in a Host_Number_Of_Completed_Packets command.

Bluetooth

4 HCI COMMANDS

4.1 INTRODUCTION

The HCI provides a uniform command method of accessing the Bluetooth hardware capabilities. The HCI Link commands provide the Host with the ability to control the link layer connections to other Bluetooth devices. These commands typically involve the Link Manager (LM) to exchange LMP commands with remote Bluetooth devices. For details see "Link Manager Protocol" on page 185.

 The HCI Policy commands are used to affect the behavior of the local and remote LM. These Policy commands provide the Host with methods of influencing how the LM manages the piconet. The Host Controller & Baseband, Informational, and Status commands provide the Host access to various registers in the Host Controller.

HCI commands may take different amounts of time to be completed. Therefore, the results of commands will be reported back to the Host in the form of an event. For example, for most HCI commands the Host Controller will generate the Command Complete event when a command is completed. This event contains the return parameters for the completed HCI command. To detect errors on the HCI-Transport Layer a response timeout needs to be defined between the Host Controller receiving a command and sending a response to the command (e.g. a Command Complete or Command Status event). Since the maximum response timeout is strongly dependent on the HCI-Transport Layer used, it is recommended to use a default value of one second for this timer. This amount of time is also dependent on the number of commands unprocessed in the command queue.

4.2 TERMINOLOGY

Baseband Packet: The smallest unit of data that is transmitted by one device to another, as defined by the "Baseband Specification" on page 33.

Packet: A higher-level protocol message than the baseband packet, currently only L2CAP (see "Logical Link Control and Adaptation Protocol Specification" on page 245) is defined, but additional packet types may be defined later.

Connection Handle: A connection handle is a 12-bit identifier which is used to uniquely address a data/voice connection from one Bluetooth device to another. The connection handles can be visualized as identifying a unique data pipe that connects two Bluetooth devices. The connection handle is maintained for the lifetime of a connection, including when a device enters Park, Sniff, or Hold mode. The Connection Handle value has local scope between Host and Host Controller. There can be multiple connection handles for any given pair of Bluetooth devices but only one ACL connection.

Event: A mechanism that the HCI uses to notify the Host for command completion, link layer status changes, etc.

Bluetooth

4.3 DATA AND PARAMETER FORMATS

- All values are in Binary and Hexadecimal Little Endian formats unless other-wise noted

- In addition, all parameters which can have negative values must use 2's complement when specifying values

- Arrayed parameters are specified using the following notation: Parame-terA[i]. If more than one set of arrayed parameters are specified (e.g. ParameterA[i], ParameterB[i]), then the order of the parameters are as fol-lows: ParameterA[0], ParameterB[0], ParameterA[1], ParameterB[1], ParameterA[2], ParameterB[2], … ParameterA[n], ParameterB[n]

- Unless noted otherwise, all parameter values are sent and received in Little Endian format (i.e. for multi-byte parameters the rightmost (Least Significa-tion Byte) is transmitted first)

- All command and event parameters that are not-arrayed and all elements in an arrayed parameter have fixed sizes (an integer number of bytes). The parameters and the size of each not arrayed parameter (or of each element in an arrayed parameter) contained in a command or an event is specified for each command or event. The number of elements in an arrayed parame-ter is not fixed.

4.4 EXCHANGE OF HCI-SPECIFIC INFORMATION

The Host Controller Transport Layer provides transparent exchange of HCI-specific information. These transporting mechanisms provide the ability for the Host to send HCI commands, ACL data, and SCO data to the Host Controller. These transport mechanisms also provide the ability for the Host to receive HCI events, ACL data, and SCO data from the Host Controller.

Since the Host Controller Transport Layer provides transparent exchange of HCI-specific information, the HCI specification specifies the format of the com-mands, events, and data exchange between the Host and the Host Controller. The next sections specify the HCI packet formats.

4.4.1 HCI Command Packet

The HCI Command Packet is used to send commands to the Host Controller from the Host. The format of the HCI Command Packet is shown in Figure 4.1, and the definition of each field is explained below. When the Host Controller completes most of the commands, a Command Complete event is sent to the Host. Some commands do not receive a Command Complete event when they have been completed. Instead, when the Host Controller receives one of these commands the Host Controller sends a Command Status event back to the Host when it has begun to execute the command. Later on, when the actions associated with the command have finished, an event that is associated with the sent command will be sent by the Host Controller to the Host. However, if the command does not begin to execute (there may be a parameter error or the

command may currently not be allowed), the event associated with the sent command will not be returned. The Command Status event will, in this case, return the appropriate error code in the Status parameter. On initial power-on, and after a reset, the Host can send a maximum of one outstanding HCI Command Packet until a Command Complete or Command Status event has been received. If an error occurs for a command for which a Command Complete event is returned, the Return_Parameters field may not contain all the return parameters specified for the command. The Status parameter, which explains the error reason and which is the first return parameter, will always be returned. If there is a Connection_Handle parameter or a BD_ADDR parameter right after the Status parameter, this parameter will also be returned so that the Host can identify to which instance of a command the Command Complete event belongs. In this case, the Connection_Handle or BD_ADDR parameter will have exactly the same value as that in the corresponding command parameter. It is implementation specific whether more parameters will be returned in case of an error.

Note: The BD_ADDR return parameter of the command Read_BD_ADDR is not used to identify to which instance of the Read_BD_ADDR command the Command Complete event belongs. It is therefore not mandatory for the Host Controller to return this parameter in case of an error.

If an error occurs for a command for which no Command Complete event is returned, all parameters returned with the event associated with this command may not be valid. The Host must take care as to which parameters may have valid values depending on the value of the Status parameter of the Complete event associated with the given command. The Command Complete and Command Status events contain a parameter called Num_HCI_Command_Packets, which indicates the number of HCI Command Packets the Host is currently allowed to send to the Host Controller. The Host Controller may buffer one or more HCI command packets, but the Host Controller must start performing the commands in the order in which they are received. The Host Controller can start performing a command before it completes previous commands. Therefore, the commands do not always complete in the order they are started. The Host Controller must be able to accept HCI Command Packets with up to 255 bytes of data excluding the HCI Command Packet header.

Each command is assigned a 2 byte Opcode used to uniquely identify different types of commands. The Opcode parameter is divided into two fields, called the OpCode Group Field (OGF) and OpCode Command Field (OCF). The OGF occupies the upper 6 bits of the Opcode, while the OCF occupies the remaining 10 bits. The OGF of 0x3F is reserved for vendor-specific debug commands. The OGF of 0x3E is reserved for Bluetooth Logo Testing. The organization of the Opcodes allows additional information to be inferred without fully decoding the entire Opcode.

Note: the OGF composed of all 'ones' has been reserved for vendor-specific debug commands. These commands are vendor-specific and are used during manufacturing, for a possible method for updating firmware, and for debugging.
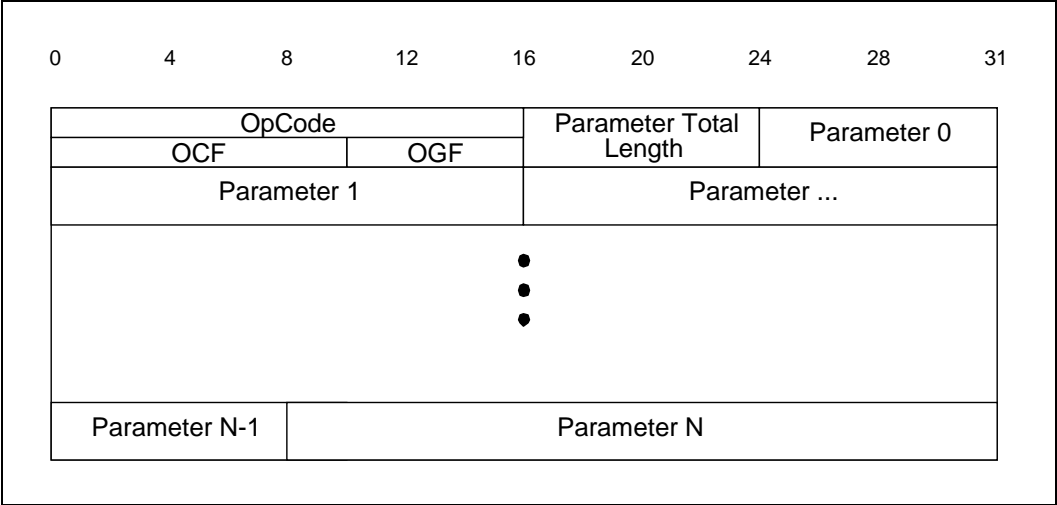
Figure 4.1: HCI Command Packet

Op_Code: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	OGFRange (6 bits): 0x00-0x3F (0x3E reserved for Bluetooth logo testing and 0x3F reserved for vendor-specific debug commands)
	OCF Range (10 bits): 0x0000-0x03FF

Parameter_Total_Length: *Size: 1 Byte*

Value	Parameter Description
0xXX	Lengths of all of the parameters contained in this packet measured in bytes. (N.B.: total length of parameters, <u>not</u> number of parameters)

Parameter 0 - N: *Size: Parameter Total Length*

Value	Parameter Description
0xXX	Each command has a specific number of parameters associated with it. These parameters and the size of each of the parameters are defined for each command. Each parameter is an integer number of bytes in size.

Bluetooth

4.4.2 HCI Event Packet

The HCI Event Packet is used by the Host Controller to notify the Host when events occur. The Host must be able to accept HCI Event Packets with up to 255 bytes of data excluding the HCI Event Packet header. The format of the HCI Event Packet is shown in Figure 4.2, and the definition of each field is explained below.
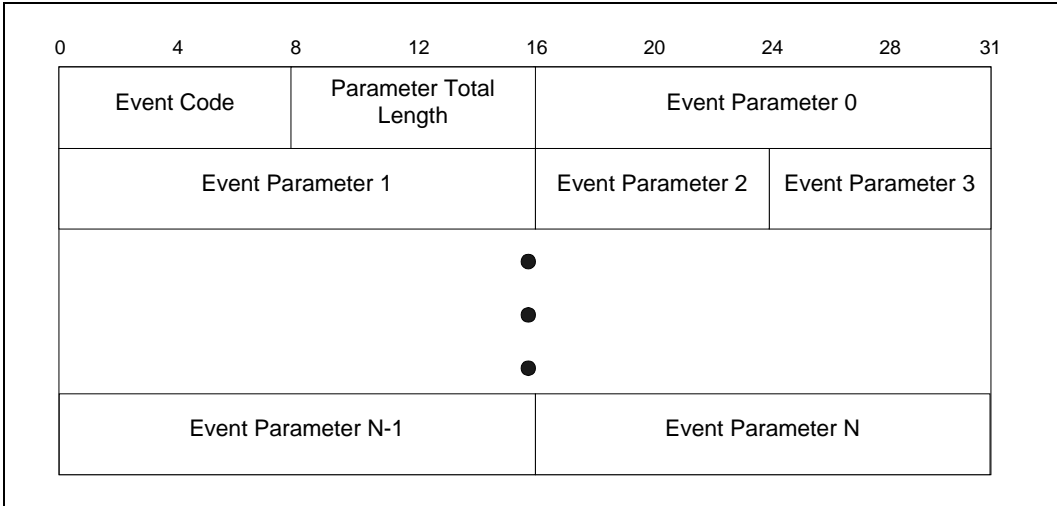
Figure 4.2: HCI Event Packet

Event_Code: *Size: 1 Byte*

Value	Parameter Description
0xXX	Each event is assigned a 1-Byte event code used to uniquely identify different types of events.
	Range: 0x00-0xFF (The event code 0xFF is reserved for the event code used for vendor-specific debug events. In addition, the event code 0xFE is also reserved for Bluetooth Logo Testing)

Parameter_Total_Length: *Size: 1 Byte*

Value	Parameter Description
0xXX	Length of all of the parameters contained in this packet, measured in bytes

Event_Parameter 0 - N: *Size: Parameter Total Length*

Value	Parameter Description
0xXX	Each event has a specific number of parameters associated with it. These parameters and the size of each of the parameters are defined for each event. Each parameter is an integer number of bytes in size.

Bluetooth

4.4.3 HCI Data Packets

HCI Data Packets are used to exchange data between the Host and Host Controller. The data packets are defined for both ACL and SCO data types. The format of the HCI ACL Data Packet is shown in Figure 4.3, and the format of the SCO Data Packet is shown in Figure 4.4. The definition for each of the fields in the data packets is explained below.

0	4	8	12	16	20	24	28	31

Connection Handle	PB Flag	BC Flag	Data Total Length
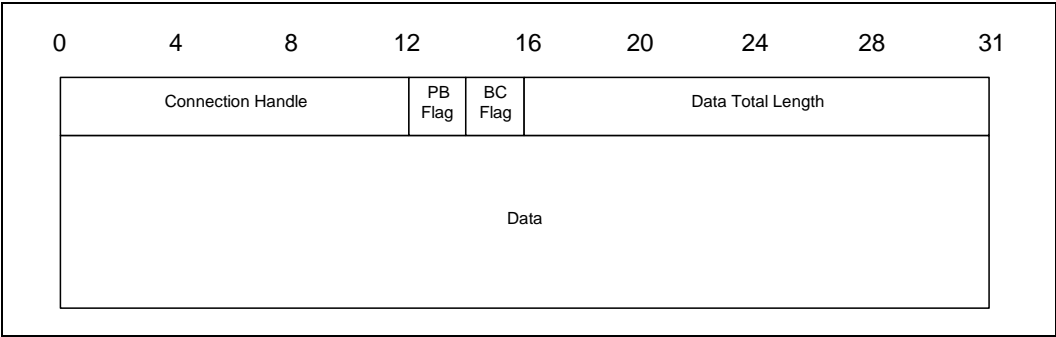
Data

Figure 4.3: HCI ACL Data Packet

Host Controller Interface Functional Specification

Bluetooth.

Connection_Handle: *Size: 12 Bits*

Value	Parameter Description
0xXXX	Connection Handle to be used for transmitting a data packet or segment.
	Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)
	The first time the Host sends an HCI Data Packet with Broadcast_Flag set to 01b (active broadcast) or 10b (piconet broadcast) after a power-on or a reset, the value of the Connection_Handle parameter must be a value which is not currently assigned by the Host Controller. The Host must use different connection handles for active broadcast and piconet broadcast. The Host Controller must then continue to use the same connection handles for each type of broadcast until a reset is made.
	Note: The Host Controller must not send a Connection Complete event containing a new Connection_Handle that it knows is used for broadcast. Note: In some situations, it may happen that the Host Controller sends a Connection Complete event before having interpreted a Broadcast packet received from the Host, and that the Connection_Handles of both Connection Complete event and HCI Data packet are the same. This conflict has to be avoided as follows:
	If a Connection Complete event is received containing one of the connection handles used for broadcast, the Host has to wait before sending any packets for the new connection until it receives a Number Of Completed Packets event indicating that there are no pending broadcast packets belonging to the connection handle. In addition, the Host must change the Connection_Handle used for the corresponding type of broadcast to a Connection_Handle which is currently not assigned by the Host Controller. This Connection_Handle must then be used for all the following broadcasts of that type until a reset is performed or the same conflict situation happens again. However, this will occur very rarely.
	The Host Controller must, in the above conflict case, be able to distinguish between the Broadcast message sent by the Host and the new connection made (this could be even a new SCO link) even though the connection handles are the same.
	For an HCI Data Packet sent from the Host Controller to the Host where the Broadcast_Flag is 01 or 10, the Connection_Handle parameter should contain the connection handle for the ACL connection to the master that sent the broadcast.
	Note: Connection handles used for Broadcast do not identify an ACL point-to-point connection, so they must not be used in any command having a Connection_Handle parameter and they will not be returned in any event having a Connection_Handle parameter except the Number Of Completed Packets event.

Flags: *Size: 2 Bits*

The Flag Bits consist of the Packet_Boundary_Flag and Broadcast_Flag. The Packet_Boundary_Flag is located in bit 4 and bit 5, and the Broadcast_Flag is located in bit 6 and 7 in the second byte of the HCI ACL Data packet.

Bluetooth

Packet_Boundary_Flag: *Size: 2 Bits*

Value	Parameter Description
00	Reserved for future use
01	Continuing fragment packet of Higher Layer Message
10	First packet of Higher Layer Message (i.e. start of an L2CAP packet)
11	Reserved for future use

Broadcast_Flag (in packet from Host to Host Controller):   *Size: 2 Bits*

Value	Parameter Description
00	No broadcast. Only point-to-point.
01	Active Broadcast: packet is sent to all active slaves.
10	Piconet Broadcast: packet is sent to all slaves, including slaves in 'Park' mode.
11	Reserved for future use.

Broadcast_Flag (in packet from Host Controller to Host):   *Size: 2 Bits*

Value	Parameter Description
00	Point-to-point
01	Packet received at an active slave (either Active Broadcast or Piconet Broadcast)
10	Packet received at a slave in 'Park' mode (Piconet Broadcast)
11	Reserved for future use.

Data_Total_Length: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	Length of data measured in bytes.

0	4	8	12	16	20	24	28	31

Connection Handle	Reserved	Data Total Length	
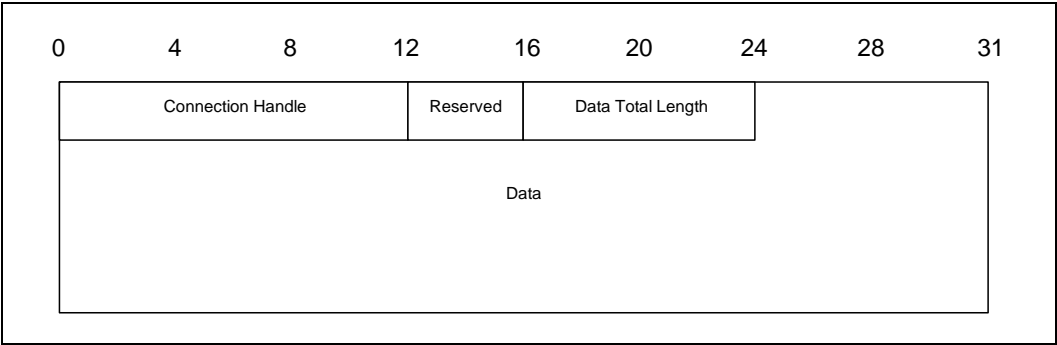
Data

Figure 4.4: HCI SCO Data Packet

Bluetooth

Connection_Handle: *Size: 12 Bits*

Value	Parameter Description
0xXXX	Connection handle to be used to for transmitting a SCO data packet or segment.
	Range: 0x0000-0x0EFF (0x0F00- 0x0FFF Reserved for future use)

The Reserved Bits consist of four bits which are located from bit 4 to bit 7 in the second byte of the HCI SCO Data packet.

Reserved: *Size: 4 Bits*

Value	Parameter Description
XXXX	Reserved for future use.

Data_Total_Length: *Size: 1 Byte*

Value	Parameter Description
0xXX	Length of SCO data measured in bytes

Bluetooth

4.5   LINK CONTROL COMMANDS

The Link Control commands allow the Host Controller to control connections to other Bluetooth devices. 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 Bluetooth devices in range, and other LMP commands. For the Link Control commands, the OGF is defined as 0x01.

Command	Command Summary Description
Inquiry	The Inquiry command will cause the Bluetooth device to enter Inquiry Mode. Inquiry Mode is used to discovery other nearby Bluetooth devices.
Inquiry_Cancel	The Inquiry_Cancel command will cause the Bluetooth device to stop the current Inquiry if the Bluetooth device is in Inquiry Mode.
Periodic_Inquiry_Mode	The Periodic_Inquiry_Mode command is used to configure the Bluetooth device to perform an automatic Inquiry based on a specified period range.
Exit_Periodic_Inquiry_Mode	The Exit_Periodic_Inquiry_Mode command is used to end the Periodic Inquiry mode when the local device is in Periodic Inquiry Mode.
Create_Connection	The Create_Connection command will cause the link manager to create an ACL connection to the Bluetooth device with the BD_ADDR specified by the command parameters.
Disconnect	The Disconnect command is used to terminate an existing connection.
Add_SCO_Connection	The Add_SCO_Connection command will cause the link manager to create a SCO connection using the ACL connection specified by the Connection Handle command parameter.
Accept_Connection_Request	The Accept_Connection_Request command is used to accept a new incoming connection request.
Reject_Connection_Request	The Reject_Connection_Request command is used to decline a new incoming connection request.
Link_Key_Request_Reply	The Link_Key_Request_Reply command is used to reply to a Link Key Request event from the Host Controller, and specifies the Link Key stored on the Host to be used as the link key for the connection with the other Bluetooth device specified by BD_ADDR.

Host Controller Interface Functional Specification

Bluetooth™

Command	Command Summary Description
Link_Key_Request_Negative_Reply	The Link_Key_Request_Negative_Reply command is used to reply to a Link Key Request event from the Host Controller if the Host does not have a stored Link Key for the connection with the other Bluetooth Device specified by BD_ADDR.
PIN_Code_Request_Reply	The PIN_Code_Request_Reply command is used to reply to a PIN Code Request event from the Host Controller and specifies the PIN code to use for a connection.
PIN_Code_Request_Negative_Reply	The PIN_Code_Request_Negative_Reply command is used to reply to a PIN Code Request event from the Host Controller when the Host cannot specify a PIN code to use for a connection.
Change_Connection_Packet_Type	The Change_Connection_Packet_Type command is used to change which packet types can be used for a connection that is currently established.
Authentication_Requested	The Authentication_Requested command is used to establish authentication between the two devices associated with the specified Connection Handle.
Set_Connection_Encryption	The Set_Connection_Encryption command is used to enable and disable the link level encryption.
Change_Connection_Link_Key	The Change_Connection_Link_Key command is used to force both devices of a connection associated to the connection handle, to generate a new link key.
Master_Link_Key	The Master_Link_Key command is used to force both devices of a connection associated to the connection handle to use the temporary link key of the Master device or the regular link keys.
Remote_Name_Request	The Remote_Name_Request command is used to obtain the user-friendly name of another Bluetooth device.
Read_Remote_Supported_Features	The Read_Remote_Supported_Features command requests a list of the supported features of a remote device.
Read_Remote_Version_Information	The Read_Remote_Version_Information command will read the values for the version information for the remote Bluetooth device.
Read_Clock_Offset	The Read_Clock_Offset command allows the Host to read the clock offset of remote devices.

Bluetooth

4.5.1 Inquiry

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

Description:

This command will cause the Bluetooth device to enter Inquiry Mode. Inquiry Mode is used to discover other nearby Bluetooth devices. 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 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. A Command Status event is sent from the Host Controller to the Host when the Inquiry command has been started by the Bluetooth device. When the Inquiry process is completed, the Host Controller will send an Inquiry Complete event to the Host indicating that the Inquiry has finished. The event parameters of Inquiry Complete event will have a summary of the result from the Inquiry process, which reports the number of nearby Bluetooth devices that responded. When a Bluetooth device responds to the Inquiry message, an Inquiry Result event will occur to notify the Host of the discovery.

A device which responds during an inquiry or inquiry period should always be reported to the Host in an 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 Set_Event_Filter. If the device has been reported earlier during the current inquiry or inquiry period, it may or may not be reported depending on the implementation (depending on if earlier results have been saved in the Host Controller and in that case how many responses that have been saved). It is recommended that the Host Controller tries to report a particular device only once during an inquiry or inquiry period.

Command Parameters:

LAP: *Size: 3 Bytes*

Value	Parameter Description
0x9E8B00– 0X9E8B3F	This is the LAP from which the inquiry access code should be derived when the inquiry procedure is made; see "Bluetooth Assigned Numbers" on page 1009.

Bluetooth.

Inquiry_Length: Size: 1 Byte

Value	Parameter Description
N = 0xXX	Maximum amount of time specified before the Inquiry is halted. Size: 1 byte Range: 0x01 – 0x30 Time = N * 1.28 sec Range: 1.28 – 61.44 Sec

Num_Responses: Size: 1 Byte

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

Return Parameters:

None.

Event(s) generated (unless masked away):

A Command Status event is sent from the Host Controller to the Host when the Host Controller has started the Inquiry process. An Inquiry Result event will be created for each Bluetooth device which responds to the Inquiry message. In addition, multiple Bluetooth devices which respond to the Inquire message may be combined into the same event. An Inquiry Complete event is generated when the Inquiry process has completed.
Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, the Inquiry Complete event will indicate that this command has been completed. No Inquiry Complete event will be generated for the canceled Inquiry process.

Bluetooth

4.5.2 Inquiry_Cancel

Command	OCF	Command Parameters	Return Parameters
HCI_Inquiry_Cancel	0x0002		Status

Description:

This command will cause the Bluetooth device to stop the current Inquiry if the Bluetooth device is in Inquiry Mode. This command allows the Host to interrupt the Bluetooth device and request the Bluetooth device to perform a different task. The command should only be issued after the Inquiry command has been issued, a Command Status event has been received for the Inquiry command, and before the Inquiry Complete event occurs.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Inquiry_Cancel command succeeded.
0x01-0xFF	Inquiry_Cancel command failed. See Table 6.1 on page 745 for list of Error Codes.

Event(s) generated (unless masked away):

When the Inquiry Cancel command has completed, a Command Complete event will be generated. No Inquiry Complete event will be generated for the canceled Inquiry process.

4.5.3 Periodic_Inquiry_Mode

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 Periodic_Inquiry_Mode command is used to configure the Bluetooth device 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 Host Controller will 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 specifies the total duration of the InquiryMode 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 Bluetooth device, and a Command Complete event is sent from the Host Controller to the Host. When each of the periodic Inquiry processes are completed, the Host Controller will send an Inquiry Complete event to the Host indicating that the latest periodic Inquiry process has finished. The event parameters of Inquiry Complete event will have a summary of the result from the previous Periodic Inquiry process, which reports the number of nearby Bluetooth devices that responded. When a Bluetooth device responds to the Inquiry message an Inquiry Result event will occur to notify the Host of the discovery.
Note: Max_Period_Length > Min_Period_Length > Inquiry_Length

A device which responds during an inquiry or inquiry period should always be reported to the Host in an 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 Set_Event_Filter. If the device has been reported earlier during the current inquiry or inquiry period, it may or may not be reported depending on the implementation (depending on if earlier results have been saved in the Host Controller and in that case how many responses that have been saved). It is recommended that the Host Controller tries to report a particular device only once during an inquiry or inquiry period.

Command Parameters:

Max_Period_Length:　　　　　　　　　　　　　　　　　　*Size: 2 Bytes*

Value	Parameter Description
N = 0xXXXX	Maximum amount of time specified between consecutive inquiries.
	Size: 2 bytes
	Range: 0x03 – 0xFFFF
	Time = N * 1.28 sec
	Range: 3.84 – 83884.8 Sec
	0.0 – 23.3 hours

Min_Period_Length:　　　　　　　　　　　　　　　　　　*Size: 2 Bytes*

Value	Parameter Description
N = 0xXXXX	Minimum amount of time specified between consecutive inquiries.
	Size: 2 bytes
	Range: 0x02 – 0xFFFE
	Time = N * 1.28 sec
	Range: 2.56 – 83883.52 Sec
	0.0 – 23.3 hours

LAP:　　　　　　　　　　　　　　　　　　　　　　　　　*Size: 3 Bytes*

Value	Parameter Description
0x9E8B00– 0X9E8B3F	This is the LAP from which the inquiry access code should be derived when the inquiry procedure is made, see "Bluetooth Assigned Numbers" on page 1009.

Inquiry_Length:　　　　　　　　　　　　　　　　　　　　*Size: 1 Byte*

Value	Parameter Description
N = 0xXX	Maximum amount of time specified before the Inquiry is halted.
	Size: 1 byte
	Range: 0x01 – 0x30
	Time = N * 1.28 sec
	Range: 1.28 – 61.44 Sec

Num_Responses:　　　　　　　　　　　　　　　　　　　　*Size: 1 Byte*

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

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Periodic Inquiry Mode command succeeded.
0x01-0xFF	Periodic Inquiry Mode command failed. See Table 6.1 on page 745 for list of Error Codes.

Event(s) generated (unless masked away):

The Periodic Inquiry Mode begins when the Host Controller sends the Command Complete event for this command to the Host. An Inquiry Result event will be created for each Bluetooth device which responds to the Inquiry message. In addition, multiple Bluetooth devices which response to the Inquiry message may be combined into the same event. An Inquiry Complete event is generated when each of the periodic Inquiry processes has completed. No Inquiry Complete event will be generated for the canceled Inquiry process.

Bluetooth

4.5.4 Exit_Periodic_Inquiry_Mode

Command	OCF	Command Parameters	Return Parameters
HCI_Exit_Periodic_Inquiry_Mode	0x0004		Status

Description:

The Exit Periodic Inquiry Mode command is used to end the Periodic Inquiry mode when the local device is in Periodic Inquiry Mode. If the local device is currently in an Inquiry process, the Inquiry process will be stopped directly and the Host Controller will no longer perform periodic inquiries until the Periodic Inquiry Mode command is reissued.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Exit Periodic Inquiry Mode command succeeded.
0x01-0xFF	Exit Periodic Inquiry Mode command failed. See Table 6.1 on page 745 for list of Error Codes.

Event(s) generated (unless masked away):

A Command Complete event for this command will occur when the local device is no longer in Periodic Inquiry Mode. No Inquiry Complete event will be generated for the canceled Inquiry process.

4.5.5 Create_Connection

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

Description:

This command will cause the Link Manager to create a connection to the Blue-tooth device with the BD_ADDR specified by the command parameters. This command causes the local Bluetooth device 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 Link Manager must use only the packet type(s) specified by the Packet_Type command parameter for sending HCI ACL Data Packets. 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 and Page_Scan_Mode parameters specify the page scan modes supported by the remote device with the BD_ADDR. This is the information that was acquired during the inquiry process. The Clock_Offset parameter is the difference between its own clock and the clock of the remote device with BD_ADDR. Only bits 2 through 16 of the difference are used, and they are mapped to this parameter as bits 0 through 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 Connection Complete event (see below). The Allow_Role_Switch parameter specifies if the local device accepts or rejects the request of a master-slave role switch when the remote device requests it at the connection setup (in the Role parameter of the Accept_Connection_Request command) (before the local Host Controller returns a Connection Complete event). For a definition of the different packet types see the "Baseband Specification" on page 33.

Note: At least one packet type must be specified. The Host should enable as many packet types as possible for the Link Manager to perform efficiently. However, the Host must not enable packet types that the local device does not support.

Command Parameters:

BD_ADDR: *Size: 6 Bytes*

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

Packet_Type: *Size: 2 Bytes*

Value	Parameter Description
0x0001	Reserved for future use.
0x0002	Reserved for future use.
0x0004	Reserved for future use.
0x0008	DM1
0x0010	DH1
0x0020	Reserved for future use.
0x0040	Reserved for future use.
0x0080	Reserved for future use.
0x0100	Reserved for future use.
0x0200	Reserved for future use.
0x0400	DM3
0x0800	DH3
0x1000	Reserved for future use.
0x2000	Reserved for future use.
0x4000	DM5
0x8000	DH5

Page_Scan_Repetition_Mode: *Size: 1 Byte*

Value	Parameter Description
0x00	R0
0x01	R1
0x02	R2
0x03 – 0xFF	Reserved.

Bluetooth

Page_Scan_Mode: *Size: 1 Byte*

Value	Parameter Description
0x00	Mandatory Page Scan Mode.
0x01	Optional Page Scan Mode I.
0x02	Optional Page Scan Mode II.
0x03	Optional Page Scan Mode III.
0x04 – 0xFF	Reserved.

Clock_Offset: *Size: 2 Bytes*

Bit format	Parameter Description
Bit 14.0	Bit 16.2 of CLKslave-CLKmaster.
Bit 15	Clock_Offset_Valid_Flag Invalid Clock Offfset = 0 Valid Clock Offset = 1

Allow_Role_Switch: *Size: 1 Byte*

Value	Parameter Description
0x00	The local device will be a master, and will not accept a master-slave switch requested by the remote device at the connection setup.
0x01	The local device may be a master, or may become a slave after accepting a master-slave switch requested by the remote device at the connection setup.
0x02-0xFF	Reserved for future use.

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Host Controller receives the Create Connection command, the Host Controller sends the Command Status event to the Host. In addition, when the LM determines the connection is established, the Host Controller, on both Bluetooth devices that form the connection, will send a Connection Complete event to each Host. The Connection Complete event contains the Connection Handle if this command is successful.
Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, the Connection Complete event will indicate that this command has been completed.

4.5.6 Disconnect

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

Description:

The Disconnection 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. The remote Bluetooth device will receive the Reason command parameter in the Disconnection Complete event. All SCO connections on a physical link should be disconnected before the ACL connection on the same physical connection is disconnected.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle for the connection being disconnected. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Reason: *Size: 1 Byte*

Value	Parameter Description
0x13-0x15, 0x1A	Other End Terminated Connection error codes (0x13-0x15) and Unsupported Remote Feature error code (0x1A) see Table 6.1 on page 745 for list of Error Codes.

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Host Controller receives the Disconnect command, it sends the Command Status event to the Host. The Disconnection Complete event will occur at each Host when the termination of the connection has completed, and indicates that this command has been completed.
Note: No Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, the Disconnection Complete event will indicate that this command has been completed.

Bluetooth

4.5.7 Add_SCO_Connection

Command	OCF	Command Parameters	Return Parameters
HCI_Add_SCO_Connection	0x0007	Connection_Handle, Packet_Type	

Description:

This command will cause the link manager to create a SCO connection using the ACL connection specified by the Connection_Handle command parameter. This command causes the local Bluetooth device to create a SCO connection. The Link Manager will determine how the new connection is established. This 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 should use for the connection. The Link Manager must only use the packet type(s) specified by the Packet_Type command parameter for sending HCI SCO Data Packets. Multiple packet types may be specified for the Packet_Type command parameter by performing a bitwise OR operation of the different packet types. The Link Manager may choose which packet type is to be used from the list of acceptable packet types. A Connection Handle for this connection is returned in the Connection Complete event (see below).
Note: SCO-Connection can only be created when an ACL Connection already exists. For a definition of the different packet types, see the "Baseband Specification" on page 33.

Note: At least one packet type must be specified. The Host should enable as many packet types as possible for the Link Manager to perform efficiently. However, the Host must not enable packet types that the local device does not support.

Command Parameters:

Connection_Handle   *Size 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle for the ACL connection being used to create an SCO connection.
	Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Bluetooth

Packet_Type: *Size: 2 Bytes*

Value	Parameter Description
0x0001	Reserved for future use.
0x0002	Reserved for future use.
0x0004	Reserved for future use.
0x0008	Reserved for future use.
0x0010	Reserved for future use.
0x0020	HV1
0x0040	HV2
0x0080	HV3
0x0100	Reserved for future use.
0x0200	Reserved for future use.
0x0400	Reserved for future use.
0x0800	Reserved for future use.
0x1000	Reserved for future use.
0x2000	Reserved for future use.
0x4000	Reserved for future use.
0x8000	Reserved for future use.

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Host Controller receives the Add_SCO_Connection command, it sends the Command Status event to the Host. In addition, when the LM determines the connection is established, the Host Controller, on both Bluetooth devices that form the connection, will send a Connection Complete event to each Host. The Connection Complete event contains the Connection Handle if this command is successful.
Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, the Connection Complete event will indicate that this command has been completed.

Bluetooth

4.5.8 Accept_Connection_Request

Command	OCF	Command Parameters	Return Parameters
HCI_Accept_Connection Request	0x0009	BD_ADDR, Role	

Description:

The Accept_Connection_Request command is used to accept a new incoming connection request. The Accept_Connection_Request command shall only be issued after a Connection Request event has occurred. The 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 Bluetooth device, 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 perform a Master-Slave switch, and become the Master for this connection. Also, the decision to accept a connection must be completed before the connection accept timeout expires on the local Bluetooth Module.
Note: when accepting SCO connection request, the Role parameter is not used and will be ignored by the Host Controller.

Command Parameters:

BD_ADDR: *Size: 6 Bytes*

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

Role: *Size: 1 Byte*

Value	Parameter Description
0x00	Become the Master for this connection. The LM will perform the Master/ Slave switch.
0x01	Remain the Slave for this connection. The LM will **NOT** perform the Master/Slave switch.

Return Parameters:

None.

Bluetooth

Event(s) generated (unless masked away):

The Accept_Connection_Request command will cause the Command Status event to be sent from the Host Controller when the Host Controller begins setting up the connection. In addition, when the Link Manager determines the connection is established, the Host Controllers on both Bluetooth devices that form the connection will send a Connection Complete event to each Host. The Connection Complete event contains the Connection Handle if this command is successful.

Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, the Connection Complete event will indicate that this command has been completed.

4.5.9 Reject_Connection_Request

Command	OCF	Command Parameters	Return Parameters
HCI_Reject_Connection _Request	0x000A	BD_ADDR, Reason	

Description:

The Reject_Connection_Request command is used to decline a new incoming connection request. The Reject_Connection_Request command shall only be called after a Connection Request event has occurred. The 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 Connection Complete event returned to the Host of the connection device, to indicate why the connection was declined.

Command Parameters:

BD_ADDR: *Size: 6 Bytes*

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

Reason: *Size: 1 Byte*

Value	Parameter Description
0x0D-0x0F	Host Reject Error Code. See Table 6.1 on page 745 for list of Error Codes and descriptions.

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Host Controller receives the Reject_Connection_Request command, the Host Controller sends the Command Status event to the Host. A Connection Complete event will then be sent, both to the local Host and to the Host of the device attempting to make the connection. The Status parameter of the 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.
Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, the Connection Complete event will indicate that this command has been completed.

Bluetooth

4.5.10 Link_Key_Request_Reply

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

Description:

The Link_Key_Request_Reply command is used to reply to a Link Key Request event from the Host Controller, and specifies the Link Key stored on the Host to be used as the link key for the connection with the other Bluetooth Device specified by BD_ADDR. The Link Key Request event will be generated when the Host Controller needs a Link Key for a connection.

When the Host Controller generates a 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 a Create_Connection or Authentication_Requested command from the remote Host), the local Host must respond with either a Link_Key_Request_Reply or Link_Key_Request_Negative_Reply command before the remote Link Manager detects LMP response timeout. (See "Link Manager Protocol" on page 185.)

Command Parameters:

BD_ADDR: *Size: 6 Bytes*

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

Link_Key: *Size: 16 Bytes*

Value	Parameter Description
0xXXXXXXXXXX XXXXXXXXXX XXXXXXXXXX	Link Key for the associated BD_ADDR.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Link_Key_Request_Reply command succeeded.
0x01-0xFF	Link_Key_Request_Reply command failed. See Table 6.1 on page 745 for list of Error Codes.

BD_ADDR: *Size: 6 Bytes*

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):

The Link_Key_Request_Reply command will cause a Command Complete event to be generated.

Bluetooth.

4.5.11 Link_Key_Request_Negative_Reply

Command	OCF	Command Parameters	Return Parameters
HCI_Link_Key_Request _Negative_Reply	0x000C	BD_ADDR	Status, BD_ADDR

Description:

The Link_Key_Request_Negative_Reply command is used to reply to a Link Key Request event from the Host Controller if the Host does not have a stored Link Key for the connection with the other Bluetooth Device specified by BD_ADDR. The Link Key Request event will be generated when the Host Controller needs a Link Key for a connection.

When the Host Controller generates a 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 a Create_Connection or Authentication_Requested command from the remote Host), the local Host must respond with either a Link_Key_Request_Reply or Link_Key_Request_Negative_Reply command before the remote Link Manager detects LMP response timeout. (See "Link Manager Protocol" on page 185.)

Command Parameters:

BD_ADDR: *Size: 6 Bytes*

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

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Link_Key_Request_Negative_Reply command succeeded.
0x01-0xFF	Link_Key_Request_Negative_Reply command failed. See Table 6.1 on page 745 for list of Error Codes.

BD_ADDR: *Size: 6 Bytes*

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

Event(s) generated (unless masked away):

The Link_Key_Request_Negative_Reply command will cause a Command Complete event to be generated.

4.5.12 PIN_Code_Request_Reply

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

Description:

The PIN_Code_Request_Reply command is used to reply to a PIN Code request event from the Host Controller, and specifies the PIN code to use for a connection. The PIN Code Request event will be generated when a connection with remote initiating device has requested pairing.

When the Host Controller generates a 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 a Create_Connection or Authentication_Requested command from the remote Host), the local Host must respond with either a PIN_Code_Request_Reply or PIN_Code_Request_Negative_Reply command before the remote Link Manager detects LMP response timeout. (See "Link Manager Protocol" on page 185.)

Command Parameters:

BD_ADDR: *Size: 6 Bytes*

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

PIN_Code_Length: *Size: 1 Byte*

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

PIN_Code: *Size: 16 Bytes*

Value	Parameter Description
0xXXXXXXXXXX XXXXXXXXXX XXXXXXXXXX	PIN code for the device that is to be connected. The Host should insure that strong PIN Codes are used. PIN Codes can be up to a maximum of 128 bits. The MSB of the PIN Code occupies byte zero.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	PIN_Code_Request_Reply command succeeded.
0x01-0xFF	PIN_Code_Request_Reply command failed. See Table 6.1 on page 745 for list of Error Codes.

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXX XXXX	BD_ADDR of the Device which the PIN Code request reply has completed.

Event(s) generated (unless masked away):

The PIN_Code_Request_Reply command will cause a Command Complete event to be generated.

Bluetooth

4.5.13 PIN_Code_Request_Negative_Reply

Command	OCF	Command Parameters	Return Parameters
HCI_PIN_Code_ Request_Negative Reply	0x000E	BD_ADDR	Status, BD_ADDR

Description:

The PIN_Code_Request_Negative_Reply command is used to reply to a PIN Code request event from the Host 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 Host Controller generates a 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 a Create_Connection or Authentication_Requested command from the remote Host), the local Host must respond with either a PIN_Code_Request_Reply or PIN_Code_Request_Negative_Reply command before the remote Link Manager detects LMP response timeout. (See "Link Manager Protocol" on page 185.)

Command Parameters:

BD_ADDR: *Size: 6 Bytes*

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

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	PIN_Code_Request_Negative_Reply command succeeded.
0x01-0xFF	PIN_Code_Request_Negative_Reply command failed. See Table 6.1 on page 7450 for list of Error Codes.

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXX XXXX	BD_ADDR of the Device which the PIN Code request negative reply has completed.

Event(s) generated (unless masked away):

The PIN_Code_Request_Negative_Reply command will cause a Command Complete event to be generated.

Bluetooth

4.5.14 Change_Connection_Packet_Type

Command	OCF	Command Parameters	Return Parameters
HCI_Change_Connection_ Packet_Type	0x000F	Connection_Handle, Packet_Type	

Description:

The 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 Link Manager must only use the packet type(s) specified by the Packet_Type command parameter for sending HCI Data Packets. The interpretation of the value for the Packet_Type command parameter will depend on the Link_Type command parameter returned in the Connection Complete event at the connection setup. Multiple packet types may be specified for the Packet_Type command parame- ter by bitwise OR operation of the different packet types. For a definition of the different packet types see the "Baseband Specification" on page 33.

Note: At least one packet type must be specified. The Host should enable as many packet types as possible for the Link Manager to perform efficiently. However, the Host must not enable packet types that the local device does not support.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to for transmitting and receiving voice or data. Returned from creating a connection. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Packet_Type: *Size: 2 Bytes*

For ACL Link_Type

Value	Parameter Description
0x0001	Reserved for future use.
0x0002	Reserved for future use.
0x0004	Reserved for future use.
0x0008	DM1

Bluetooth

Value	Parameter Description
0x0010	DH1
0x0020	Reserved for future use.
0x0040	Reserved for future use.
0x0080	Reserved for future use.
0x0100	Reserved for future use.
0x0200	Reserved for future use.
0x0400	DM3
0x0800	DH3
0x1000	Reserved for future use.
0x2000	Reserved for future use.
0x4000	DM5
0x8000	DH5

For SCO Link Type

Value	Parameter Description
0x0001	Reserved for future use.
0x0002	Reserved for future use.
0x0004	Reserved for future use.
0x0008	Reserved for future use.
0x0010	Reserved for future use.
0x0020	HV1
0x0040	HV2
0x0080	HV3
0x0100	Reserved for future use.
0x0200	Reserved for future use.
0x0400	Reserved for future use.
0x0800	Reserved for future use.
0x1000	Reserved for future use.
0x2000	Reserved for future use.
0x4000	Reserved for future use.
0x8000	Reserved for future use.

Bluetooth

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Host Controller receives the Change Connection Packet Type command, the Host Controller sends the Command Status event to the Host. In addition, when the Link Manager determines the packet type has been changed for the connection, the Host Controller on the local device will send a Connection Packet Type Changed event to the Host. This will be done at the local side only.
Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, the Connection Packet Type Changed event will indicate that this command has been completed.

Bluetooth

4.5.15 Authentication_Requested

Command	OCF	Command Parameters	Return Parameters
HCI_Authentication_ Requested	0x0011	Connection_Handle	

Description:

The Authentication_Requested command is used to try to authenticate the remote device associated with the specified Connection Handle. The Host must not issue the Authentication_Requested command with a Connection_Handle corresponding to an encrypted link. On an authentication failure, the Host Controller or Link Manager shall not automatically detach the link. The Host is responsible for issuing a Disconnect command to terminate the link if the action is appropriate.

Note**:** the Connection_Handle command parameter is used to identify the other Bluetooth device, which forms the connection. The Connection Handle should be a Connection Handle for an ACL connection.

Command Parameters:

Connection_Handle: *Size 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to set up authentication for all Connection Handles with the same Bluetooth device end-point as the specified Connection Handle.
	Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Host Controller receives the Authentication_Requested command, it sends the Command Status event to the Host. The Authentication Complete event will occur when the authentication has been completed for the connection and is indication that this command has been completed.
Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, the Authentication Complete event will indicate that this command has been completed.

Note: When the local or remote Host Controller does not have a link key for the specified Connection_Handle, it will request the link key from its Host, before the local Host finally receives the Authentication Complete event.

Bluetooth

4.5.16 Set_Connection_Encryption

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

Description:

The Set_Connection_Encryption command is used to enable and disable the link level encryption. Note**:** the Connection_Handle command parameter is used to identify the other Bluetooth device which forms the connection. The Connection Handle should be a Connection Handle for an ACL connection. While the encryption is being changed, all ACL traffic must be turned off for all Connection Handles associated with the remote device.

Command Parameters:

Connection_Handle: *Size 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to enable/disable the link layer encryption for all Connection Handles with the same Bluetooth device end-point as the specified Connection Handle.
	Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Encryption_Enable: *Size: 1 Byte*

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 Host Controller receives the Set_Connection_Encryption command, the Host Controller sends the Command Status event to the Host. When the Link Manager has completed enabling/disabling encryption for the connection, the Host Controller on the local Bluetooth device will send an Encryption Change event to the Host, and the Host Controller on the remote device will also generate an Encryption Change event.
Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, the Encryption Change event will indicate that this command has been completed.

4.5.17 Change_Connection_Link_Key

Command	OCF	Command Parameters	Return Parameters
HCI_Change_Connection_ Link_Key	0x0015	Connection_Handle	

Description:

The 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 Bluetooth device forming the connection. The Connection Handle should be a Connection Handle for an ACL connection. If the connection encryption is enabled, and the temporary link key is currently used, then the Bluetooth master device will automatically restart the encryption.

Command Parameters:

Connection_Handle: *Size 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle used to identify a connection. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Host Controller receives the Change_Connection_Link_Key command, the Host Controller sends the Command Status event to the Host. When the Link Manager has changed the Link Key for the connection, the Host Controller on the local Bluetooth device will send a Link Key Notification event and a Change Connection Link Key Complete event to the Host, and the Host Controller on the remote device will also generate a Link Key Notification event. The Link Key Notification event indicates that a new connection link key is valid for the connection.
Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, the Change Connection Link Key Complete event will indicate that this command has been completed.

Bluetooth

4.5.18 Master_Link_Key

Command	OCF	Command Parameters	Return Parameters
HCI_Master_Link_Key	0x0017	Key_Flag	

Description:

The Master Link Key command is used to force the device that is master of the piconet to use the temporary link key of the master device, 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 of the Master, or the semi-permanent link keys) shall be used.

Command Parameters:

Key_Flag: *Size: 1 Byte*

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 Host Controller receives the Master_Link_Key command, the Host Controller sends the Command Status event to the Host. When the Link Manager has changed link key, the Host Controller on both the local and the remote device will send a Master Link Key Complete event to the Host. The Connection Handle on the master side will be a Connection Handle for one of the existing connections to a slave. On the slave side, the Connection Handle will be a Connection Handle to the initiating master.

The Master Link Key Complete event contains the status of this command. Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, the Master Link Key Complete event will indicate that this command has been completed.

4.5.19 Remote_Name_Request

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

Description:

The Remote_Name_Request command is used to obtain the user-friendly name of another Bluetooth device. The user-friendly name is used to enable the user to distinguish one Bluetooth device 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 and Page_Scan_Mode command parameters specify the page scan modes supported by the remote device with the BD_ADDR. This is the information that was acquired during the inquiry process. The Clock_Offset parameter is the difference between its own clock and the clock of the remote device with BD_ADDR. Only bits 2 through 16 of the difference are used and they are mapped to this parameter as bits 0 through 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.
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 name of the remote device.

Command Parameters:

BD_ADDR: *Size: 6 Bytes*

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

Page_Scan_Repetition_Mode: *Size: 1 Byte*

Value	Parameter Description
0x00	R0
0x01	R1
0x02	R2
0x03 – 0xFF	Reserved.

Bluetooth.

Page_Scan_Mode: *Size: 1 Byte*

Value	Parameter Description
0x00	Mandatory Page Scan Mode.
0x01	Optional Page Scan Mode I.
0x02	Optional Page Scan Mode II.
0x03	Optional Page Scan Mode III.
0x04 – 0xFF	Reserved.

Clock_Offset: *Size: 2 Bytes*

Bit format	Parameter Description
Bit 14.0	Bit 16.2 of CLKslave-CLKmaster.
Bit 15	Clock_Offset_Valid_Flag Invalid Clock Offfset = 0 Valid Clock Offset = 1

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Host Controller receives the Remote_Name_Request command, the Host Controller sends the Command Status event to the Host. When the Link Manager has completed the LMP messages to obtain the remote name, the Host Controller on the local Bluetooth device will send a Remote Name Request Complete event to the Host. Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, only the Remote Name Request Complete event will indicate that this command has been completed.

Bluetooth

4.5.20 Read_Remote_Supported_Features

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 must be a Connection_Handle for an ACL connection. The Read Remote Supported Features Complete event will return a list of the LMP features. For details see "Link Manager Protocol" on page 185.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's LMP-supported features list to get.
	Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Host Controller receives the Read_Remote_Supported_Features command, the Host Controller sends the Command Status event to the Host. When the Link Manager has completed the LMP messages to determine the remote features, the Host Controller on the local Bluetooth device will send a Read Remote Supported Features Complete event to the Host. The Read Remote Supported Features Complete event contains the status of this command, and parameters describing the supported features of the remote device. Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, the Read Remote Supported Features Complete event will indicate that this command has been completed.

Bluetooth

4.5.21 Read_Remote_Version_Information

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 Bluetooth device identified by the Connection_Handle parameter. The Connection_Handle must be a Connection_Handle for an ACL connection.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's version information to get. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Host Controller receives the Read_Remote_Version_Information command, the Host Controller sends the Command Status event to the Host. When the Link Manager has completed the LMP messages to determine the remote version information, the Host Controller on the local Bluetooth device will send a Read Remote Version Information Complete event to the Host. The Read Remote Version Information Complete event contains the status of this command, and parameters describing the version and subversion of the LMP used by the remote device.
Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, the Read Remote Version Information Complete event will indicate that this command has been completed.

4.5.22 Read_Clock_Offset

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Clock_Offset	0x001F	Connection_Handle	

Description:

Both the System Clock and the clock offset of 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 clock offset of remote devices. The Connection_Handle must be a Connection_Handle for an ACL connection. This command could be used to facilitate handoffs of Bluetooth devices from one device to another.

Command Parameters:

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

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's Clock Offset parameter is returned. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Host Controller receives the Read_Clock_Offset command, the Host Controller sends the Command Status event to the Host. If this command was requested at the master and the Link Manager has completed the LMP messages to obtain the Clock Offset information, the Host Controller on the local Bluetooth device will send a Read Clock Offset Complete event to the Host. Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, only the Read Clock Offset Complete event will indicate that this command has been completed. If the command is requested at the slave, the LM will immediately send a Command Status event and a Read Clock Offset Complete event to the Host, without an exchange of LMP PDU.

Bluetooth

4.6   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 main-
tained, 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 Bluetooth Devices, and
therefore there can only be one ACL HCI Connection Handle for each physical
link layer Connection. The Bluetooth Host Controller provides policy adjust-
ment 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 Bluetooth devices. For the Link Policy Commands, the OGF
is defined as 0x02.

Command	Command Summary Description
Hold_Mode	The Hold_Mode command is used to alter the behavior of the LM and have the LM place the local or remote device into the hold mode.
Sniff_Mode	The Sniff_Mode command is used to alter the behavior of the LM and have the LM place the local or remote device into the sniff mode.
Exit_Sniff_Mode	The Exit_Sniff_Mode command is used to end the sniff mode for a connection handle which is currently in sniff mode.
Park_Mode	The Park_Mode command is used to alter the behavior of the LM and have the LM place the local or remote device into the Park mode.
Exit_Park_Mode	The Exit_Park_Mode command is used to switch the Bluetooth device from park mode back to active mode.
QoS_Setup	The QoS_Setup command is used to specify Quality of Service parameters for a connection handle.
Role_Discovery	The Role_Discovery command is used for a Bluetooth device to determine which role the device is performing for a particular Connection Handle.
Switch_Role	The Switch_Role command is used for a Bluetooth device switch the current role the device is performing for a particular connection with the specified Bluetooth device

Bluetooth.

Command	Command Summary Description
Read_Link_Policy_Settings	The Read_Link_Policy_Settings command will read the Link Policy settings for the specified Connection Handle. The Link Policy settings allow the Host to specify which Link Modes the LM can use for the specified Connection Handle.
Write_Link_Policy_Settings	The Write_Link_Policy_Settings command will write the Link Policy settings for the specified Connection Handle. The Link Policy settings allow the Host to specify which Link Modes the LM can use for the specified Connection Handle.

Bluetooth

4.6.1 Hold_Mode

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

Description:

The 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 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 cannot be greater than the Hold_Mode_Max_Interval. The Host Controller will return the actual Hold interval in the Interval parameter of the Mode Change event, if the command is successful. This command enables the Host to sup- port a low-power policy for itself or several other Bluetooth devices, and allows the devices to enter Inquiry Scan, Page Scan, and a number of other possible actions.
Note: the connection handle cannot be of the SCO link type.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify a connection. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Hold_Mode_Max_Interval: *Size: 2 Bytes*

Value	Parameter Description
N = 0xXXXX	Maximum acceptable number of Baseband slots to wait in Hold Mode. Time Length of the Hold = N * 0.625 msec (1 Baseband slot) Range for N: 0x0001-0xFFFF Time Range: 0.625ms - 40.9 sec

Bluetooth

Hold_Mode_Min_Interval: *Size: 2 Bytes*

Value	Parameter Description
N = 0xXXXX	Minimum acceptable number of Baseband slots to wait in Hold Mode.
	Time Length of the Hold = N * 0.625 msec (1 Baseband slot)
	Range for N: 0x0001-0xFFFF
	Time Range: 0.625 msec - 40.9 sec

Return Parameters:

None**.**

Event(s) generated (unless masked away):

The Host Controller sends the Command Status event for this command to the Host when it has received the Hold_Mode command. The Mode Change event will occur when the Hold Mode has started and the Mode Change event will occur again when the Hold Mode has completed for the specified connection handle. The 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 Bluetooth device. Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, only the Mode Change event will indicate that this command has been completed. If an error occurs after the Command Status event has occurred, then the status in the Mode Change event will indicate the error.

Bluetooth

4.6.2 Sniff_Mode

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

Description:

The 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 the sniff mode. The Connection_Handle command parameter is used to identify which ACL link connection is to be placed in sniff mode. The Sniff_Max_Interval and Sniff_Min_Interval command parameters are used to specify the requested acceptable maximum and minimum periods in the Sniff Mode. The Sniff_Min_Interval cannot be greater than the Sniff_Max_Interval. The sniff interval defines the amount of time between each consecutive sniff period. The Host Controller will return the actual sniff interval in the Interval parameter of the Mode Change event, if the command is successful. The slave will listen at the end of every actual sniff interval, for the period specified by the Sniff_Attempt command parameter. The slave will continue listening for packets for an additional period specified by Sniff_Timeout, as long as it is receiving packets. This command enables the Host to support a low-power policy for itself or several other Bluetooth devices, and allows the devices to enter Inquiry Scan, Page Scan, and a number of other possible actions.
Note: in addition, the connection handle cannot be one of SCO link type.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify a connection. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Sniff_Max_Interval: *Size: 2 Byte*

Value	Parameter Description
N = 0xXXXX	Maximum acceptable number of Baseband slots between each sniff period. (Sniff_Max_Interval >= Sniff_Min_Interval) Length = N * 0.625 msec (1 Baseband slot) Range for N: 0x0001 – 0xFFFF Time Range: 0.625 msec - 40.9 Seconds

Bluetooth

Sniff_Min_Interval: *Size: 2 Byte*

Value	Parameter Description
N = 0xXXXX	Minimum acceptable number of Baseband slots between each sniff period.
	(Sniff_Max_Interval >= Sniff_Min_Interval)
	Length = N * 0.625 msec (1 Baseband slot)
	Range for N: 0x0001 – 0xFFFF
	Time Range: 0.625 msec - 40.9 Seconds

Sniff_Attempt: *Size: 2 Byte*

Value	Parameter Description
N = 0xXXXX	Number of Baseband slots for sniff attempt.
	Length = N * 0.625 msec (1 Baseband slot)
	Range for N: 0x0001 – 0xFFFF
	Time Range: 0.625 msec - 40.9 Seconds

Sniff_Timeout: *Size: 2 Byte*

Value	Parameter Description
N = 0xXXXX	Number of Baseband slots for sniff timeout.
	Length = N * 0.625 msec (1 Baseband slot)
	Range for N: 0x0001 – 0xFFFF
	Time Range: 0.625 msec - 40.9 Seconds

Return Parameters:

None.

Event(s) generated (unless masked away):

The Host Controller sends the Command Status event for this command to the Host when it has received the Sniff_Mode command. The Mode Change event will occur when the Sniff Mode has started for the specified connection handle. Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead only the Mode Change event will indicate that this command has been completed. If an error occurs after the Command Status event has occurred, then the status in the Mode Change event will indicate the error.

Bluetooth

4.6.3 Exit_Sniff_Mode

Command	OCF	Command Parameters	Return Parameters
HCI_Exit_Sniff_Mode	0x0004	Connection_Handle	

Description:

The Exit_Sniff_Mode command is used to end the sniff mode for a connection handle, which is currently in sniff mode. The Link Manager will determine and issue the appropriate LMP commands to remove the sniff mode for the associated Connection Handle.
Note: in addition, the connection handle cannot be one of SCO link type.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify a connection.
	Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

None.

Event(s) generated (unless masked away):

A Command Status event for this command will occur when Host Controller has received the Exit_Sniff_Mode command. The Mode Change event will occur when the Sniff Mode has ended for the specified connection handle. Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, only the Mode Change event will indicate that this command has been completed.

Bluetooth

4.6.4 Park_Mode

Command	OCF	Command Parameters	Return Parameters
HCI_Park_Mode	0x0005	Connection_Handle, Beacon_Max_Interval, Beacon_Min_Interval	

Description:

The Park Mode command is used to alter the behavior of the Link Manager, and have the LM place the baseband connection associated by the specified Connection Handle into the Park mode. The Connection_Handle command parameter is used to identify which connection is to be placed in Park mode. The Connection_Handle must be a Connection_Handle for an ACL connection. The Beacon Interval command parameters specify the acceptable length of the interval between beacons. However, the remote device may request shorter interval. The Beacon_Max_Interval parameter specifies the acceptable longest length of the interval between beacons. The Beacon_Min_Interval parameter specifies the acceptable shortest length of the interval between beacons. Therefore, the Beacon Min Interval cannot be greater than the Beacon Max Interval. The Host Controller will return the actual Beacon interval in the Interval parameter of the Mode Change event, if the command is successful. This command enables the Host to support a low-power policy for itself or several other Bluetooth devices, allows the devices to enter Inquiry Scan, Page Scan, provides support for large number of Bluetooth Devices in a single piconet, and a number of other possible activities.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify a connection. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Beacon_Max_Interval: *Size: 2 Bytes*

Value	Parameter Description
N = 0xXXXX	Maximum acceptable number of Baseband slots between consecutive beacons. Interval Length = N * 0.625 msec (1 Baseband slot) Range for N: 0x0001 – 0xFFFF Time Range: 0.625 msec - 40.9 Seconds

Bluetooth

Beacon_Min_Interval   *Size: 2 Bytes*

Value	Parameter Description
N = 0xXXXX	Minimum acceptable number of Baseband slots between consecutive beacons
	Interval Length = N * 0.625 msec (1 Baseband slot)
	Range for N: 0x0001 – 0xFFFF
	Time Range: 0.625 msec - 40.9 Seconds

Return Parameters:

None.

Event(s) generated (unless masked away):

The Host Controller sends the Command Status event for this command to the Host when it has received the Park_Mode command. The Mode Change event will occur when the Park Mode has started for the specified connection handle. Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, only the Mode Change event will indicate that this command has been completed. If an error occurs after the Command Status event has occurred, then the status in the Mode Change event will indicate the error.

Bluetooth

4.6.5 Exit_Park_Mode

Command	OCF	Command Parameters	Return Parameters
HCI_Exit_Park_Mode	0x0006	Connection_Handle	

Description:

The Exit_Park_Mode command is used to switch the Bluetooth device from park mode back to active mode. This command may only be issued when the device associated with the specified Connection Handle is in Park Mode. The Connection_Handle must be a Connection_Handle for an ACL connection. This function does not complete immediately.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify a connection.
	Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

None.

Event(s) generated (unless masked away):

A Command Status event for this command will occur when the Host Controller has received the Exit_Park_Mode command. The Mode Change event will occur when the Park Mode has ended for the specified connection handle. Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, only the Mode Change event will indicate that this command has been completed.

Bluetooth.

4.6.6 QoS_Setup

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

Description:

The QoS_Setup command is used to specify Quality of Service parameters for a connection handle. The Connection_Handle must be a Connection_Handle for an ACL connection. These QoS parameter are the same parameters as L2CAP QoS. For more detail see "Logical Link Control and Adaptation Protocol Specification" on page 245. 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. Bluetooth devices that are both slaves and masters can use this command. When a device is a slave, this command will trigger an LMP request to the master to provide the slave with the specified QoS as determined by the LM. When a device is a master, this command is used to request a slave device to accept the specified QoS as determined by the LM of the master. The Connection_Handle command parameter is used to identify for which connection the QoS request is requested.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify which connection for the QoS Setup. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Flags: *Size: 1 Byte*

Value	Parameter Description
0x00 – 0xFF	Reserved for Future Use.

Bluetooth

Service_Type: *Size: 1 Byte*

Value	Parameter Description
0x00	No Traffic.
0x01	Best Effort.
0x02	Guaranteed.
0x03-0xFF	Reserved for Future Use.

Token_Rate: *Size: 4 Bytes*

Value	Parameter Description
0xXXXXXXXX	Token Rate in bytes per second.

Peak_Bandwidth: *Size: 4 Bytes*

Value	Parameter Description
0xXXXXXXXX	Peak Bandwidth in bytes per second.

Latency: *Size: 4 Bytes*

Value	Parameter Description
0xXXXXXXXX	Latency in microseconds.

Delay_Variation: *Size: 4 Bytes*

Value	Parameter Description
0xXXXXXXXX	Delay Variation in microseconds.

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Host Controller receives the QoS_Setup command, the Host Controller sends the Command Status event to the Host. When the Link Manager has completed the LMP messages to establish the requested QoS parameters, the Host Controller on the local Bluetooth device will send a 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 QoS Setup Complete event may, however, be different on the initiating and the remote side. The QoS Setup Complete event returned by the Host Controller on the local side contains the status of this command, and returned QoS parameters describing the supported QoS for the connection.
Note: No Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, the QoS Setup Complete event will indicate that this command has been completed.

Bluetooth

4.6.7 Role_Discovery

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

Description:

The Role_Discovery command is used for a Bluetooth device to determine which role the device is performing for a particular Connection Handle. The Connection_Handle must be a Connection_Handle for an ACL connection.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify a connection. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Role_Discovery command succeeded,
0x01-0xFF	Role_Discovery command failed. See Table 6.1 on page 745 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify which connection the Role_Discovery command was issued on. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Current_Role: *Size: 1 Byte*

Value	Parameter Description
0x00	Current Role is Master for this Connection Handle.
0x01	Current Role is Slave for this Connection Handle.

Event(s) generated (unless masked away):

When the Role_Discovery command has completed, a Command Complete event will be generated.

4.6.8 Switch_Role

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

Description:

The Switch_Role command is used for a Bluetooth device to switch the current role the device is performing for a particular connection with another specified Bluetooth device. The BD_ADDR command parameter indicates for which connection the role switch is to be performed. The Role indicates the requested new role that the local device performs.
Note: the BD_ADDR command parameter must specify a Bluetooth device for which a connection already exists.

Command Parameters:

BD_ADDR: *Size: 6 Bytes*

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

Role: *Size: 1 Byte*

Value	Parameter Description
0x00	Change own Role to Master for this BD_ADDR.
0x01	Change own Role to Slave for this BD_ADDR.

Return Parameters:

None.

Event(s) generated (unless masked away):

A Command Status event for this command will occur when the Host Controller has received the Switch_Role command. When the role switch is performed, a Role Change event will occur to indicate that the roles have been changed, and will be communicated to both Hosts.
Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, only the Role Change event will indicate that this command has been completed.

Bluetooth

4.6.9 Read_Link_Policy_Settings

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 Link_Policy_Settings parameter determines the behavior of the local Link Manager when it receives a request from a remote device or it determines itself to change the master-slave role or to enter the hold, sniff, or park mode. The local Link Manager will automatically accept or reject such a request from the remote device, and may even autonomously request itself, depending on the value of the Link_Policy_Settings parameter for the corresponding Connection_Handle. When the value of the Link_Policy_Settings parameter is changed for a certain Connection_Handle, the new value will only be used for requests from a remote device or from the local Link Manager itself made after this command has been completed. The Connection_Handle must be a Connection_Handle for an ACL connection. By enabling each mode individually, the Host can choose any combination needed to support various modes of operation. Multiple LM policies may be specified for the Link_Policy_Settings parameter by performing a bitwise OR operation of the different activity types.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify a connection. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Link_Policy_Settings command succeeded.
0x01-0xFF	Read_Link_Policy_Settings command failed. See Table 6.1 on page 745 for list of Error Codes.

Bluetooth

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify a connection.
	Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Link_Policy_Settings   *Size: 2 Bytes*

Value	Parameter Description
0x0000	Disable All LM Modes.
0x0001	Enable Master Slave Switch.
0x0002	Enable Hold Mode.
0x0004	Enable Sniff Mode.
0x0008	Enable Park Mode.
0x0010 – 0x8000	Reserved for Future Use.

Event(s) generated (unless masked away):

When the Read_Link_Policy_Settings command has completed, a Command Complete event will be generated.

Bluetooth

4.6.10 Write_Link_Policy_Settings

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Link_Policy_ Settings	0x000D	Connection_Handle, Link_Policy_Settings	Status, Connection_Handle

Description:

This command will write the Link Policy setting for the specified Connection Handle. The Link_Policy_Settings parameter determines the behavior of the local Link Manager when it receives a request from a remote device or it determines itself to change the master-slave role or to enter the hold, sniff, or park mode. The local Link Manager will automatically accept or reject such a request from the remote device, and may even autonomously request itself, depending on the value of the Link_Policy_Settings parameter for the corresponding Connection_Handle. When the value of the Link_Policy_Settings parameter is changed for a certain Connection_Handle, the new value will only be used for requests from a remote device or from the local Link Manager itself made after this command has been completed. The Connection_Handle must be a Connection_Handle for an ACL connection. By enabling each mode individually, the Host can choose any combination needed to support various modes of operation. Multiple LM policies may be specified for the Link_Policy_Settings parameter by performing a bitwise OR operation of the different activity types.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify a connection. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Bluetooth

Link_Policy_Settings							*Size: 2 Bytes*

Value	Parameter Description
0x0000	Disable All LM Modes **Default**.
0x0001	Enable Master Slave Switch.
0x0002	Enable Hold Mode.
0x0004	Enable Sniff Mode.
0x0008	Enable Park Mode.
0x0010 – 0x8000	Reserved for Future Use.

Return Parameters:

Status:								*Size: 1 Byte*

Value	Parameter Description
0x00	Write_Link_Policy_Settings command succeeded.
0x01-0xFF	Write_Link_Policy_Settings command failed. See Table 6.1 on page 745 for list of Error Codes.

Connection_Handle:				*Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify a connection. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Event(s) generated (unless masked away):

When the Write_Link_Policy_Settings command has completed, a Command Complete event will be generated.

Bluetooth

4.7   HOST CONTROLLER & BASEBAND COMMANDS

The Host Controller & Baseband Commands provide access and control to various capabilities of the Bluetooth hardware. These parameters provide control of Bluetooth devices and of the capabilities of the Host Controller, Link Manager, and Baseband. The host device can use these commands to modify the behavior of the local device. For the HCI Control and Baseband Commands, the OGF is defined as 0x03

Command	Command Summary Description
Set_Event_Mask	The Set_Event_Mask command is used to control which events are generated by the HCI for the Host.
Reset	The Reset command will reset the Bluetooth Host Controller, Link Manager, and the radio module.
Set_Event_Filter	The 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.
Flush	The Flush command is used to discard all data that is currently pending for transmission in the Host Controller for the specified connection handle, even if there currently are chunks of data that belong to more than one L2CAP packet in the Host Controller.
Read_PIN_Type	The Read_PIN_Type command is used for the Host to read the value that is specified to indicate whether the Host supports variable PIN or only fixed PINs.
Write_PIN_Type	The Write_PIN_Type command is used for the Host to specify whether the Host supports variable PIN or only fixed PINs.
Create_New_Unit_Key	The Create_New_Unit_Key command is used to create a new unit key.
Read_Stored_Link_Key	The Read_Stored_Link_Key command provides the ability to read one or more link keys stored in the Bluetooth Host Controller.
Write_Stored_Link_Key	The Write_Stored_Link_Key command provides the ability to write one or more link keys to be stored in the Bluetooth Host Controller.

Command	Command Summary Description
Delete_Stored_Link_Key	The Delete_Stored_Link_Key command provides the ability to remove one or more of the link keys stored in the Bluetooth Host Controller.
Change_Local_Name	The Change_Local_Name command provides the ability to modify the user-friendly name for the Bluetooth device.
Read_Local_Name	The Read_Local_Name command provides the ability to read the stored user-friendly name for the Bluetooth device.
Read_Connection_Accept_Timeout	The Read_Connection_Accept_Timeout command will read the value for the Connection_Accept_Timeout configuration parameter, which allows the Bluetooth hardware to automatically deny a connection request after a specified period has occurred, and to refuse a new connection.
Write_Connection_Accept_Timeout	The Write_Connection_Accept_Timeout will write the value for the Connection_Accept_Timeout configuration parameter, which allows the Bluetooth hardware to automatically deny a connection request after a specified period has occurred, and to refuse a new connection.
Read_Page_Timeout	The Read_Page_Timeout command will read the value for the Page_Reply_Timeout configuration parameter, which allows the Bluetooth hardware to define the amount of time a connection request will wait for the remote device to respond before the local device returns a connection failure.
Write_Page_Timeout	The Write_Page_Timeout command will write the value for the Page_Reply_Timeout configuration parameter, which allows the Bluetooth hardware to define the amount of time a connection request will wait for the remote device to respond before the local device returns a connection failure.
Read_Scan_Enable	The Read_Scan_Enable command will read the value for the Scan_Enable configuration parameter, which controls whether or not the Bluetooth device will periodically scan for page attempts and/or inquiry requests from other Bluetooth devices.

Bluetooth

Command	Command Summary Description
Write_Scan_Enable	The Write_Scan_Enable command will write the value for the Scan_Enable configuration parameter, which controls whether or not the Bluetooth device will periodically scan for page attempts and/or inquiry requests from other Bluetooth devices.
Read_Page_Scan_Activity	The Read_Page_Scan_Activity command will read the values for the Page_Scan_Interval and Page_Scan_Window configuration parameters. Page_Scan_Interval defines the amount of time between consecutive page scans. Page_Scan_Window defines the duration of the page scan.
Write_Page_Scan_Activity	The Write_Page_Scan_Activity command will write the value for Page_Scan_Interval and_ Page_Scan_Window configuration parameters. Page_Scan_Interval defines the amount of time between consecutive page scans. Page_Scan_Window defines the duration of the page scan.
Read_Inquiry_Scan_Activity	The Read_Inquiry_Scan_Activity command will read the value for Inquiry_Scan_Interval and Inquiry_Scan_Window configuration parameters. Inquiry_Scan_Interval defines the amount of time between consecutive inquiry scans. Inquiry_Scan_Window defines the amount of time for the duration of the inquiry scan.
Write_Inquiry_Scan_Activity	The Write_Inquiry_Scan_Activity command will write the value for Inquiry_Scan_Interval and Inquiry_Scan_Window configuration parameters. Inquiry_Scan_Interval defines the amount of time between consecutive inquiry scans. Inquiry_Scan_Window defines the amount of time for the duration of the inquiry scan.
Read_Authentication_Enable	The Read_Authentication_Enable command will read the value for the Authentication_ Enable parameter, which controls whether the Bluetooth device will require authentication for each connection with other Bluetooth devices.
Write_Authentication_Enable	The Write_Authentication_Enable command will write the value for the Authentication_ Enable parameter, which controls whether the Bluetooth device will require authentication for each connection with other Bluetooth devices.
Read_Encryption_Mode	The Read_Encryption_Mode command will read the value for the Encryption_Mode parameter, which controls whether the Bluetooth device will require encryption for each connection with other Bluetooth devices.

Command	Command Summary Description
Write_Encryption_Mode	The Write_Encryption_Mode command will write the value for the Encryption_Mode parameter, which controls whether the Bluetooth device will require encryption for each connection with other Bluetooth devices.
Read_Class_of_Device	The Read_Class_of_Device command will read the value for the Class_of_Device parameter, which is used to indicate its capabilities to other devices.
Write_Class_of_Device	The Write_Class_of_Device command will write the value for the Class_of_Device parameter, which is used to indicate its capabilities to other devices.
Read_Voice_Setting	The Read_Voice_Setting command will read the values for the Voice_Setting parameter, which controls all the various settings for the voice connections.
Write_Voice_Setting	The Write_Voice_Setting command will write the values for the Voice_Setting parameter, which controls all the various settings for the voice connections.
Read_Automatic_Flush_Timeout	The Read_Automatic_Flush_Timeout will read the value for the Flush_Timeout parameter for the specified connection handle. The Flush_Timeout parameter is only used for ACL connections.
Write_Automatic_Flush_Timeout	The Write_Automatic_Flush_Timeout will write the value for the Flush_Timeout parameter for the specified connection handle. The Flush_Timeout parameter is only used for ACL connections.
Read_Num_Broadcast_Retransmissions	The Read_Num_Broadcast_Retransmissions command will read the parameter value for the Number of Broadcast Retransmissions for the device. Broadcast packets are not acknowledged and are unreliable. This parameter is used to increase the reliability of a broadcast message by retransmitting the broadcast message multiple times.
Write_Num_Broadcast_Retransmissions	The Write_Num_Broadcast_Retransmissions command will write the parameter value for the Number of Broadcast Retransmissions for the device. Broadcast packets are not acknowledged and are unreliable. This parameter is used to increase the reliability of a broadcast message by retransmitting the broadcast message multiple times.

Command	Command Summary Description
Read_Hold_Mode_Activity	The Read_Hold_Mode_Activity command will read the value for the Hold_Mode_Activity parameter. This value is used to determine what activity the device should do when it is in hold mode.
Write_Hold_Mode_Activity	The Write_Hold_Mode_Activity command will write the value for the Hold_Mode_Activity parameter. This value is used to determine what activity the device should do when it is in hold mode.
Read_Transmit_Power_Level	The Read_Transmit_Power_Level command will read the values for the Transmit_Power_Level parameter for the specified Connection Handle.
Read_SCO_Flow_Control_Enable	The Read_SCO_Flow_Control_Enable command provides the ability to read the SCO_Flow_Control_Enable setting. By using this setting, the Host can decide if the Host Controller will send Number Of Completed Packets events for SCO Connection Handles.
Write_SCO_Flow_Control_Enable	The Write_SCO_Flow_Control_Enable command provides the ability to write the SCO_Flow_Control_Enable setting. By using this setting, the Host can decide if the Host Controller will send Number Of Completed Packets events for SCO Connection Handles.
Set_Host_Controller_To_Host_Flow_Control	The Set_Host_Controller_To_Host_Flow_Control command is used by the Host to turn flow control on or off in the direction from the Host Controller to the Host.
Host_Buffer_Size	The Host_Buffer_Size command is used by the Host to notify the Host Controller about its buffer sizes for ACL and SCO data. The Host Controller will segment the data to be transmitted from the Host Controller to the Host, so that data contained in HCI Data Packets will not exceed these sizes.
Host_Number_Of_Completed_Packets	The Host_Number_Of_Completed_Packets command is used by the Host to indicate to the Host Controller when the Host is ready to receive more HCI packets for any connection handle.
Read_Link_Supervision_Timeout	The Read_Link_Supervision_Timeout command will read the value for the Link_Supervision_Timeout parameter for the device. This parameter is used by the master or slave Bluetooth device to monitor link loss.

Command	Command Summary Description
Write Link_Supervision_Timeout	The Write Link_Supervision_Timeout command will write the value for the Link_Supervision_ Timeout parameter for the device. This parameter is used by the master or slave Bluetooth device to monitor link loss.
Read_Number_Of_Supported_IAC	The Read_Number_Of_Supported_IAC command will read the value for the number of Inquiry Access Codes (IAC) that the local Bluetooth device can simultaneously listen for during an Inquiry Scan.
Read_Current_IAC_LAP	The Read_Current_IAC_LAP command will read the LAP(s) used to create the Inquiry Access Codes (IAC) that the local Bluetooth device is simultaneously scanning for during Inquiry Scans.
Write_Current_IAC_LAP	The Write_Current_IAC_LAP will write the LAP(s) used to create the Inquiry Access Codes (IAC) that the local Bluetooth device is simultaneously scanning for during Inquiry Scans.
Read_Page_Scan_Period_Mode	The Read_Page_Scan_Period_Mode command is used to read the mandatory Page_Scan_Period_Mode of the local Bluetooth device.
Write_Page_Scan_Period_Mode	The Write_Page_Scan_Period_Mode command is used to write the mandatory Page_Scan_Period_Mode of the local Bluetooth device.
Read_Page_Scan_Mode	The Read_Page_Scan_Mode command is used to read the default Page_Scan_Mode of the local Bluetooth device.
Write_Page_Scan_Mode	The Write_Page_Scan_Mode command is used to write the default Page_Scan_Mode of the local Bluetooth device.

Bluetooth

4.7.1 Set_Event_Mask

Command	OCF	Command Parameters	Return Parameters
HCI_Set_Event_Mask	0x0001	Event_Mask	Status

Description:

The 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. The Host has to deal with each event that occurs by the Bluetooth devices. The event mask allows the Host to control how much it is interrupted.
Note: the Command Complete event, Command Status event and Number Of Completed Packets event cannot be masked. These events always occur. The Event_Mask is a bit mask of all of the events specified in .

Command Parameters:

Event_Mask: *Size: 8 Bytes*

Value	Parameter Description
0x0000000000000000	No events specified
0x0000000000000001	Inquiry Complete event
0x0000000000000002	Inquiry Result event
0x0000000000000004	Connection Complete event
0x0000000000000008	Connection Request event
0x0000000000000010	Disconnection Complete event
0x0000000000000020	Authentication Complete event
0x0000000000000040	Remote Name Request Complete event
0x0000000000000080	Encryption Change event
0x0000000000000100	Change Connection Link Key Complete event
0x0000000000000200	Master Link Key Complete event
0x0000000000000400	Read Remote Supported Features Complete event
0x0000000000000800	Read Remote Version Information Complete event
0x0000000000001000	QoS Setup Complete event
0x0000000000002000	Command Complete event
0x0000000000004000	Command Status event

0x0000000000008000	Hardware Error event
0x0000000000010000	Flush Occurred event
0x0000000000020000	Role Change event

Value	Parameter Description
0x0000000000040000	Number Of Completed Packets event
0x0000000000080000	Mode Change event
0x0000000000100000	Return Link Keys event
0x0000000000200000	PIN Code Request event
0x0000000000400000	Link Key Request event
0x0000000000800000	Link Key Notification event
0x0000000001000000	Loopback Command event
0x0000000002000000	Data Buffer Overflow event
0x0000000004000000	Max Slots Change event
0x0000000008000000	Read Clock Offset Complete event
0x0000000010000000	Connection Packet Type Changed event
0x0000000020000000	QoS Violation event
0x0000000040000000	Page Scan Mode Change event
0x0000000080000000	Page Scan Repetition Mode Change event
0x0000000100000000 to 0x8000000000000000	Reserved for future use
0x00000000FFFFFFFF	**Default** (All events enabled)

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Set_Event_Mask command succeeded.
0x01-0xFF	Set_Event_Mask command failed. See Table 6.1 on page 745 for list of Error Codes.

Event(s) generated (unless masked away):

When the Set_Event_Mask command has completed, a Command Complete event will be generated.

4.7.2 Reset

Command	OCF	Command Parameters	Return Parameters
HCI_Reset	0x0003		Status

Description:

The Reset command will reset the Bluetooth Host Controller, Link Manager, and the radio module. The current operational state will be lost, and all queued packets will be lost. After the reset is completed, the Bluetooth device will enter standby mode.
Note: after the reset has completed, the Host Controller will automatically revert to the default values for the parameters for which default values are defined in this specification.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Reset command succeeded, was received and will be executed.
0x01-0xFF	Reset command failed. See Table 6.1 on page 745 for list of Error Codes.

Event(s) generated (unless masked away):

Before the Reset command will be executed, a Command Complete event needs to be returned to indicate to the Host that the Reset command was received and will be executed.

4.7.3 Set_Event_Filter

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

Description:

The 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 Host 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 Host Controller to filter out Inquiry Result events. The Inquiry Result filter allows the Host to specify that the Host Controller only sends Inquiry Results to the Host if the Inquiry Result event 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. A new device responded to 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 Inquiry and Periodic Inquiry command. The Connection Setup filter allows the Host to specify that the Host Controller only sends a Connection Complete or 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

Bluetooth

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 Host Controller will send the Connection Complete event to the Host when the connection is completed) or if the Host should make the decision (in which case the Host Controller will send the Connection Request event to the Host, to elicit a decision on the connection).

The Connection Setup filter is used in conjunction with the 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 on the conditions set by the Host, and the Auto_Accept_Flag is off for this device, then a Connection Request event will be sent to the Host by the Host Controller. A 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 a Connection Complete event will be sent to the Host by the Host Controller. (No Connection Request event will be sent in that case.)

The Host Controller will store these filters in volatile memory until the Host clears the event filters using the Set_Event_Filter command or until the Reset command is issued. The number of event filters the Host Controller can store is implementation dependent. If the Host tries to set more filters than the Host Controller can store, the Host Controller will return the "Memory Full" error code 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, a Link Key Request event and possibly also a PIN Code Request event and a Link Key Notification event could be sent to the Host by the Host Controller before the 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 Byte*

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 bytes. Filter_Type should be the only parameter.)
0x01	Inquiry Result.
0x02	Connection Setup.
0x03-0xFF	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 Byte*

Value	Parameter Description
0x00	A new device responded to the Inquiry process. (Note: A device may be reported to the Host in an Inquiry Result event more than once during an inquiry or inquiry period depending on the implementation, see description in Section 4.5.1 on page 542 and Section 4.5.3 on page 545)
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.
0x03-0xFF	Reserved for Future Use

Connection_Setup_Filter_Condition_Type: *Size: 1 Byte*

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.
0x03-0xFF	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 Byte*

Value	Parameter Description
	The Condition parameter is not used.

Bluetooth.

Condition for Inquiry_Result_Filter_Condition_Type = 0x01

Condition:

Size: 6 Bytes

Class_of_Device: *Size: 3 Bytes*

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

Class_of_Device_Mask: *Size: 3 Bytes*

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 Bytes

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device of Interest

Condition for Connection_Setup_Filter_Condition_Type = 0x00

Condition:

Size: 1 Byte

Auto_Accept_Flag: *Size:1 Byte*

Value	Parameter Description
0x01	Do NOT Auto accept the connection.
0x02	Do Auto accept the connection.
0x03 – 0xFF	Reserved for future use.

Bluetooth

Condition for Connection_Setup_Filter_Condition_Type = 0x01

Condition:

Size: 7 Bytes

Class_of_Device: *Size: 3 Bytes*

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

Class_of_Device_Mask: *Size: 3 Bytes*

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.

Auto_Accept_Flag: *Size: 1 Byte*

Value	Parameter Description
0x01	Do NOT Auto accept the connection.
0x02	Do Auto accept the connection.
0x03 – 0xFF	Reserved for future use.

Condition for Connection_Setup_Filter_Condition_Type = 0x02

Condition:

Size: 7 Bytes

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device of Interest.

Auto_Accept_Flag: *Size: 1 Byte*

Value	Parameter Description
0x01	Do NOT Auto accept the connection.
0x02	Do Auto accept the connection.
0x03 – 0xFF	Reserved for future use.

Bluetooth

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Set_Event_Filter command succeeded.
0x01-0xFF	Set_Event_Filter command failed. See Table 6.1 on page 745 for list of Error Codes.

Event(s) generated (unless masked away):

A Command Complete event for this command will occur when the Host Controller has enabled the filtering of events. When one of the conditions are met, a specific event will occur.

Bluetooth

4.7.4 Flush

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

Description:

The Flush command is used to discard all data that is currently pending for transmission in the Host Controller for the specified connection handle, even if there currently are chunks of data that belong to more than one L2CAP packet in the Host Controller. After this, all data that is sent to the Host Controller for the same connection handle will be discarded by the Host Controller until an HCI Data Packet with the start Packet_Boundary_Flag (0x02) is received. When this happens, a new transmission attempt can be made. This command 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 Host Controller should be flushed. Note that the Flush command is used for ACL connections ONLY. In addition to the Flush command, the automatic flush timers (see) can be used to automatically flush the L2CAP packet that is currently being transmitted after the specified flush timer has expired.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify which connection to flush. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Flush command succeeded.
0x01-0xFF	Flush command failed. See Table 6.1 on page 745 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify which connection the flush command was issued on. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Bluetooth

Event(s) generated (unless masked away):

The Flush Occurred event will occur once the flush is completed. A Flush Occurred event could be from an automatic Flush or could be cause by the Host issuing the Flush command. When the Flush command has completed, a Command Complete event will be generated, to indicate that the Host caused the Flush.

Bluetooth

4.7.5 Read_PIN_Type

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

Description:

The Read_PIN_Type command is used for the Host to read whether the Link Manager assumes that the Host supports variable PIN codes only a fixed PIN code. The Bluetooth hardware uses the PIN-type information during pairing.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_PIN_Type command succeeded.
0x01-0xFF	Read_PIN_Type command failed. See Table 6.1 on page 745 for list of Error Codes.

PIN_Type: *Size: 1 Byte*

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

Event(s) generated (unless masked away):

When the Read_PIN_Type command has completed, a Command Complete event will be generated.

Bluetooth

4.7.6 Write_PIN_Type

Command	OCF	Command Parameters	Return Parameters
HCI_Write_PIN_Type	0x000A	PIN_Type	Status

Description:

The Write_PIN_Type command is used for the Host to write to the Host Controller whether the Host supports variable PIN codes or only a fixed PIN code. The Bluetooth hardware uses the PIN-type information during pairing.

Command Parameters:

PIN_Type: *Size: 1 Byte*

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

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write PIN Type command succeeded.
0x01-0xFF	Write PIN Type command failed. See Table 6.1 on page 745 for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_PIN_Type command has completed, a Command Complete event will be generated.

Bluetooth

4.7.7 Create_New_Unit_Key

Command	OCF	Command Parameters	Return Parameters
HCI_Create_New_Unit_Key	0x000B		Status

Description:

The Create_New_Unit_Key command is used to create a new unit key. The Bluetooth hardware will generate a random seed that will be used to generate the new unit key. All new connection will use the new unit key, but the old unit key will still be used for all current connections.
Note: this command will not have any effect for a device which doesn't use unit keys (i.e. a device which uses only combination keys).

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Create New Unit Key command succeeded.
0x01-0xFF	Create New Unit Key command failed. See Table 6.1 on page 745 for list of Error Codes.

Event(s) generated (unless masked away):

When the Create_New_Unit_Key command has completed, a Command Complete event will be generated.

Bluetooth

4.7.8 Read_Stored_Link_Key

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Stored_ Link_Key	0x000D	BD_ADDR, Read_All_Flag	Status, Max_Num_Keys, Num_Keys_Read

Description:

The Read_Stored_Link_Key command provides the ability to read one or more link keys stored in the Bluetooth Host Controller. The Bluetooth Host Controller can store a limited number of link keys for other Bluetooth devices. Link keys are shared between two Bluetooth devices, and are used for all security trans-actions 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 Bluetooth Host Controller when needed. The Read_All_Flag parameter is used to indicate if all of the stored Link Keys should be returned. If Read_All_Flag indicates that all Link Keys are to be returned, then the BD_ADDR command parameter must 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 Return Link Keys events.

Command Parameters:

BD_ADDR: *Size: 6 Bytes*

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

Read_All_Flag: *Size: 1 Byte*

Value	Parameter Description
0x00	Return Link Key for specified BD_ADDR.
0x01	Return all stored Link Keys.
0x02-0xFF	Reserved for future use.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Stored_Link_Key command succeeded.
0x01-0xFF	Read_Stored_Link_Key command failed. See Table 6.1 on page 745 for list of Error Codes.

Bluetooth

Max_Num_Keys: *Size: 2 Byte*

Value	Parameter Description
0xXXXX	Maximum Number of Link Keys which the Host Controller can store. Range: 0x0000 – 0xFFFF

Num_Keys_Read: *Size: 2 Bytes*

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

Event(s) generated (unless masked away):

Zero or more instances of the Return Link Keys event will occur after the command is issued. When there are no link keys stored, no Return Link Keys events will be returned. When there are link keys stored, the number of link keys returned in each Return Link Keys event is implementation specific. When the Read Stored Link Key command has completed a Command Complete event will be generated.

Bluetooth

4.7.9 Write_Stored_Link_Key

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 Write_Stored_Link_Key command provides the ability to write one or more link keys to be stored in the Bluetooth Host Controller. The Bluetooth Host Controller can store a limited number of link keys for other Bluetooth devices. If no additional space is available in the Bluetooth Host 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 Host Controller. Link keys are shared between two Bluetooth devices 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 Bluetooth Host Controller when needed.
Note: Link Keys are only stored by issuing this command.

Command Parameters:

Num_Keys_To_Write: *Size: 1 Byte*

Value	Parameter Description
0xXX	Number of Link Keys to Write.

BD_ADDR [i]:   *Size: 6 Bytes * Num_Keys_To_Write*

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

Link_Key[i]:   *Size: 16 Bytes * Num_Keys_To_Write*

Value	Parameter Description
0xXXXXXXXXXX XXXXXXXXXX XXXXXXXXXX	Link Key for the associated BD_ADDR.

Return Parameters:

Status: Size: 1 Byte

Value	Parameter Description
0x00	Write_Stored_Link_Key command succeeded.
0x01-0xFF	Write_Stored_Link_Key command failed. See Table 6.1 on page 745 for list of Error Codes.

Num_Keys_Written: Size: 1 Bytes

Value	Parameter Description
0xXX	Number of Link Keys successfully written. Range: 0x00 – 0xFF

Event(s) generated (unless masked away):

When the Write_Stored_Link_Key command has completed, a Command Complete event will be generated.

4.7.10 Delete_Stored_Link_Key

Command	OCF	Command Parameters	Return Parameters
HCI_Delete_Stored_ Link_Key	0x0012	BD_ADDR, Delete_All_Flag	Status, Num_Keys_Deleted

Description:

The Delete_Stored_Link_Key command provides the ability to remove one or more of the link keys stored in the Bluetooth Host Controller. The Bluetooth Host Controller can store a limited number of link keys for other Bluetooth devices. Link keys are shared between two Bluetooth devices and are used for all security transactions between the two devices. The Delete_All_Flag parameter is used to indicate if all of the stored Link Keys should be deleted. If the Delete_All_Flag indicates that all Link Keys are to be deleted, then the BD_ADDR command parameter must 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.

Command Parameters:

BD_ADDR: *Size: 6 Bytes*

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

Delete_All_Flag: *Size: 1 Byte*

Value	Parameter Description
0x00	Delete only the Link Key for specified BD_ADDR.
0x01	Delete all stored Link Keys.
0x02-0xFF	Reserved for future use.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Delete_Stored_Link_Key command succeeded.
0x01-0xFF	Delete_Stored_Link_Key command failed. See Table 6.1 on page 745 for list of Error Codes.

Bluetooth

Num_Keys_Deleted: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	Number of Link Keys Deleted

Event(s) generated (unless masked away):

When the Delete_Stored_Link_Key command has completed, a Command
Complete event will be generated.

Bluetooth

4.7.11 Change_Local_Name

Command	OCF	Command Parameters	Return Parameters
HCI_Change_Local_Name	0x0013	Name	Status

Description:

The Change_Local_Name command provides the ability to modify the user-friendly name for the Bluetooth device. A Bluetooth device may send a request to get the user-friendly name of another Bluetooth device. The user-friendly name provides the user with the ability to distinguish one Bluetooth device from another. The Name command parameter is a UTF-8 encoded string with up to 248 bytes in length. The Name command parameter should be null-terminated (0x00) if the UTF-8 encoded string is less than 248 bytes.
Note: the Name Parameter is transmitted starting with the first byte of the name. This is an exception to the Little Endian order format for transmitting multi-byte parameters.

Command Parameters:

Name: *Size: 248 Bytes*

Value	Parameter Description
	A UTF-8 encoded User-Friendly Descriptive Name for the device. The UTF-8 encoded Name can be up to 248 bytes in length. If it is shorter than 248 bytes, the end is indicated by a NULL byte (0x00).
	Null terminated Zero length String. **Default.**

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Change_Local_Name command succeeded.
0x01-0xFF	Change_Local_Name command failed. See Table 6.1 on page 745 for list of Error Codes.

Event(s) generated (unless masked away):

When the Change_Local_Name command has completed, a Command Complete event will be generated.

Bluetooth

4.7.12 Read_Local_Name

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

Description:

The Read_Local_Name command provides the ability to read the stored user-friendly name for the Bluetooth device. The user-friendly name provides the user the ability to distinguish one Bluetooth device from another. The Name return parameter is a UTF-8 encoded string with up to 248 bytes in length. The Name return parameter will be null terminated (0x00) if the UTF-8 encoded string is less than 248 bytes.
Note: the Name Parameter is transmitted starting with the first byte of the name. This is an exception to the Little Endian order format for transmitting multi-byte parameters.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Local_Name command succeeded
0x01-0xFF	Read_Local_Name command failed see Table 6.1 on page 746 for list of Error Codes

Name: *Size: 248 Bytes*

Value	Parameter Description
	A UTF-8 encoded User Friendly Descriptive Name for the device.
	The UTF-8 encoded Name can be up to 248 bytes in length. If it is shorter than 248 bytes, the end is indicated by a NULL byte (0x00).

Event(s) generated (unless masked away):

When the Read_Local_Name command has completed a Command Complete event will be generated.

4.7.13 Read_Connection_Accept_Timeout

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Connection_ Accept_Timeout	0x0015		Status, Conn_Accept_Timeout

Description:

This command will read the value for the Connection_Accept_Timeout configuration parameter. The Connection_Accept_Timeout configuration parameter allows the Bluetooth hardware to automatically deny a connection request after a specified time period has occurred and the new connection is not accepted. The parameter defines the time duration from when the Host Controller sends a Connection Request event until the Host Controller will automatically reject an incoming connection.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Connection_Accept_Timeout command succeeded.
0x01-0xFF	Read_Connection_Accept_Timeout command failed. See Table 6.1 on page 745 for list of Error Codes.

Conn_Accept_Timeout: *Size: 2 Bytes*

Value	Parameter Description
N = 0xXXXX	Connection Accept Timeout measured in Number of Baseband slots. Interval Length = N * 0.625 msec (1 Baseband slot) Range for N: 0x0001 – 0xB540 Time Range: 0.625 msec -29 seconds

Event(s) generated (unless masked away):

When the Read_Connection_Timeout command has completed, a Command Complete event will be generated.

4.7.14 Write_Connection_Accept_Timeout

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Connection_ Accept_Timeout	0x0016	Conn_Accept_Timeout	Status

Description:

This command will write the value for the Connection_Accept_Timeout configuration parameter. The Connection_Accept_Timeout configuration parameter allows the Bluetooth hardware to automatically deny a connection request after a specified time interval has occurred and the new connection is not accepted. The parameter defines the time duration from when the Host Controller sends a Connection Request event until the Host Controller will automatically reject an incoming connection.

Command Parameters:

Conn_Accept_Timeout: *Size: 2 Bytes*

Value	Parameter Description
N = 0xXXXX	Connection Accept Timeout measured in Number of Baseband slots.
	Interval Length = N * 0.625 msec (1 Baseband slot)
	Range for N: 0x0001 – 0xB540
	Time Range: 0.625 msec - 29 seconds
	Default: N = 0x1FA0 Time = 5 Sec

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write_Connection_Accept_Timeout command succeeded.
0x01-0xFF	Write_Connection_Accept_Timeout command failed. See Table 6.1 on page 745 for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Connection_Accept_Timeout command has completed, a Command Complete event will be generated.

4.7.15 Read_Page_Timeout

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

Description:

This command will read the value for the Page_Timeout configuration parameter. The Page_Timeout configuration parameter defines the maximum time the local Link Manager will wait for a baseband page response from the remote device at a locally initiated connection attempt. If this time expires and the remote device has not responded to the page at baseband level, the connection attempt will be considered to have failed.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Page_Timeout command succeeded.
0x01-0xFF	Read_Page_Timeout command failed. See Table 6.1 on page 745 for list of Error Codes.

Page_Timeout: *Size: 2 Bytes*

Value	Parameter Description
N = 0xXXXX	Page Timeout measured in Number of Baseband slots. Interval Length = N * 0.625 msec (1 Baseband slot) Range for N: 0x0001 – 0xFFFF Time Range: 0.625 msec -40.9 Seconds

Event(s) generated (unless masked away):

When the Read_Page_Timeout command has completed, a Command Complete event will be generated.

Bluetooth

4.7.16 Write_Page_Timeout

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Page_Timeout	0x0018	Page_Timeout	Status

Description:

This command will write the value for the Page_Timeout configuration parameter. The Page_Timeout configuration parameter defines the maximum time the local Link Manager will wait for a baseband page response from the remote device at a locally initiated connection attempt. If this time expires and the remote device has not responded to the page at baseband level, the connection attempt will be considered to have failed.

Command Parameters:

Page_Timeout: *Size: 2 Bytes*

Value	Parameter Description
0	Illegal Page Timeout. Must be larger than 0.
N = 0xXXXX	Page Timeout measured in Number of Baseband slots. Interval Length = N * 0.625 msec (1 Baseband slot) Range for N: 0x0001 – 0xFFFF Time Range: 0.625 msec -40.9 Seconds **Default: N = 0x2000 Time = 5.12 Sec**

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write_Page_Timeout command succeeded.
0x01-0xFF	Write_Page_Timeout command failed. See Table 6.1 on page 745 for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Page_Timeout command has completed, a Command Complete event will be generated.

Bluetooth

4.7.17 Read_Scan_Enable

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

Description:

This command will read the value for the Scan_Enable parameter. The Scan_Enable parameter controls whether or not the Bluetooth device will peri-odically scan for page attempts and/or inquiry requests from other Bluetooth devices. If Page_Scan is enabled, then the device will enter page scan mode based on the value of the Page_Scan_Interval and Page_Scan_Window parameters. If Inquiry_Scan is enabled, then the device will enter Inquiry Scan mode based on the value of the Inquiry_Scan_Interval and Inquiry_Scan_ Window parameters.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Scan_Enable command succeeded.
0x01-0xFF	Read_Scan_Enable command failed. See Table 6.1 on page 745 for list of Error Codes.

Scan_Enable: *Size: 1 Byte*

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.

Event(s) generated (unless masked away):

When the Read_Scan_Enable command has completed, a Command Com-plete event will be generated.

Bluetooth

4.7.18 Write_Scan_Enable

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Scan_Enable	0x001A	Scan_Enable	Status

Description:

This command will write the value for the Scan_Enable parameter. The Scan_Enable parameter controls whether or not the Bluetooth device will periodically scan for page attempts and/or inquiry requests from other Bluetooth devices. If Page_Scan is enabled, then the device will enter page scan mode based on the value of the Page_Scan_Interval and Page_Scan_Window parameters. If Inquiry_Scan is enabled, then the device will enter Inquiry Scan mode based on the value of the Inquiry_Scan_Interval and Inquiry_Scan_ Window parameters.

Command Parameters:

Scan_Enable: *Size: 1 Byte*

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.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write_Scan_Enable command succeeded.
0x01-0xFF	Write_Scan_Enable command failed. See Table 6.1 on page 745 for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Scan_Enable command has completed, a Command Complete event will be generated.

4.7.19 Read_Page_Scan_Activity

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Page_Scan_ Activity	0x001B		Status, Page_Scan_Interval, Page_Scan_Window

Description:

This command will read the value for Page_Scan_Activity configuration param-eters. The Page_Scan_Interval configuration parameter defines the amount of time between consecutive page scans. This time interval is defined from when the Host Controller started its last page scan until it begins the next page scan. The Page_Scan_Window configuration parameter defines the amount of time for the duration of the page scan. The Page_Scan_Window can only be less than or equal to the Page_Scan_Interval.
Note: Page Scan is only performed when Page_Scan is enabled (see 4.7.17 and 4.7.18).

A changed Page_Scan_Interval could change the local Page_Scan_ Repetition_Mode (see "Baseband Specification" on page 33, Keyword: SR-Mode).

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Page_Scan_Activity command succeeded.
0x01-0xFF	Read_Page_Scan_Activity command failed. See Table 6.1 on page 745 for list of Error Codes.

Page_Scan_Interval: *Size: 2 Bytes*

Value	Parameter Description
N = 0xXXXX	Size: 2 Bytes Range: 0x0012 – 0x1000 Time = N * 0.625 msec Range: 11.25 msec – 2560 msec

Bluetooth

Page_Scan_Window: *Size: 2 Bytes*

Value	Parameter Description
N = 0xXXXX	Size: 2 Bytes Range: 0x0012 – 0x1000 Time = N * 0.625 msec Range: 11.25 msec – 2560 msec

Event(s) generated (unless masked away):

When the Read_Page_Scan_Activity command has completed, a Command Complete event will be generated.

4.7.20 Write_Page_Scan_Activity

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Page_Scan_ Activity	0x001C	Page_Scan_Interval, Page_Scan_Window	Status

Description:

This command will write the value for Page_Scan_Activity configuration parameter. The Page_Scan_Interval configuration parameter defines the amount of time between consecutive page scans. This is defined as the time interval from when the Host Controller started its last page scan until it begins the next page scan. The Page_Scan_Window configuration parameter defines the amount of time for the duration of the page scan. The Page_Scan_Window can only be less than or equal to the Page_Scan_Interval.
Note: Page Scan is only performed when Page_Scan is enabled (see 4.7.17 and 4.7.18). A changed Page_Scan_Interval could change the local Page_ Scan_Repetition_Mode (see "Baseband Specification" on page 33, Keyword: SR-Mode).

Command Parameters:

Page_Scan_Interval: *Size: 2 Bytes*

Value	Parameter Description
N = 0xXXXX	Size: 2 Bytes Range: 0x0012 – 0x1000 Time = N * 0.625 msec Range: 11.25 msec –2560 msec Default: N = 0x0800 Time = 1.28 Sec

Page_Scan_Window: *Size: 2 Bytes*

Value	Parameter Description
N = 0xXXXX	Size: 2 Bytes Range: 0x0012 – 0x1000 Time = N * 0.625 msec Range: 11.25 msec – 2560 msec Default: N = 0x0012 Time = 11.25 msec

Bluetooth

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write_Page_Scan_Activity command succeeded.
0x01-0xFF	Write_Page_Scan_Activity command failed. See Table 6.1 on page 745 for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Page_Scan_Activity command has completed, a Command Complete event will be generated.

4.7.21 Read_Inquiry_Scan_Activity

Command	OCF	Command Parameters	Return Parameters
HCI_Read_ Inquiry_Scan_Activity	0x001D		Status, Inquiry_Scan_Interval, Inquiry_Scan_Window

Description:

This command will read the value for Inquiry_Scan_Activity configuration parameter. The Inquiry_Scan_Interval configuration parameter defines the amount of time between consecutive inquiry scans. This is defined as the time interval from when the Host Controller started its last inquiry scan until it begins the next inquiry scan.
The Inquiry_Scan_Window configuration parameter defines the amount of time for the duration of the inquiry scan. The Inquiry_Scan_Window can only be less than or equal to the Inquiry_Scan_Interval.
Note: Inquiry Scan is only performed when Inquiry_Scan is enabled see 4.7.17 and 4.7.18).

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Inquiry_Scan_Activity command succeeded.
0x01-0xFF	Read_Inquiry_Scan_Activity command failed. See Table 6.1 on page 745 for list of Error Codes.

Inquiry_Scan_Interval: *Size: 2 Bytes*

Value	Parameter Description
N = 0xXXXX	Size: 2 Bytes Range: 0x0012 – 0x1000 Time = N * 0.625 msec Range: 11.25 – 2560 msec

Bluetooth

Inquiry_Scan_Window: *Size: 2 Bytes*

Value	Parameter Description
N = 0xXXXX	Size: 2 Bytes Range: 0x0012 – 0x1000 Time = N * 0.625 msec Range: 0.625 msec – 2560 msec

Event(s) generated (unless masked away):

When the Read_Inquiry_Scan_Activity command has completed, a Command
Complete event will be generated.

4.7.22 Write_Inquiry_Scan_Activity

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Inquiry_ Scan_Activity	0x001E	Inquiry_Scan_Interval, Inquiry_Scan_Window	Status

Description:

This command will write the value for Inquiry_Scan_Activity configuration parameter. The Inquiry_Scan_Interval configuration parameter defines the amount of time between consecutive inquiry scans. This is defined as the time interval from when the Host Controller started its last inquiry scan until it begins the next inquiry scan.

The Inquiry_Scan_Window configuration parameter defines the amount of time for the duration of the inquiry scan. The Inquiry_Scan_Window can only be less than or equal to the Inquiry_Scan_Interval.

Note: Inquiry Scan is only performed when Inquiry_Scan is enabled (see 4.7.17 and 4.7.18).

Command Parameters:

Inquiry_Scan_Interval: *Size: 2 Bytes*

Value	Parameter Description
N = 0xXXXX	Size: 2 Bytes Range: 0x0012 – 0x1000 Time = N * 0.625 msec Range: 11.25 – 2560 msec Default: N = 0x0800 Time = 1.28 Sec

Inquiry_Scan_Window: *Size: 2 Bytes*

Value	Parameter Description
N = 0xXXXX	Size: 2 Bytes Range: 0x0012 – 0x1000 Time = N * 0.625 msec Range: 11.25 msec – 2560 msec Default: N = 0x0012 Time = 11.25 msec

Bluetooth.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write_Inquiry_Scan_Activity command succeeded.
0x01-0xFF	Write_Inquiry_Scan_Activity command failed. See Table 6.1 on page 745 for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Inquiry_Scan_Activity command has completed, a Command Complete event will be generated.

4.7.23 Read_Authentication_Enable

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

Description:

This command will read the value for the Authentication_Enable parameter. The Authentication_Enable parameter controls if the local device requires to authenticate the remote device at connection setup (between the Create_Connection command or acceptance of an incoming ACL connection and the corresponding Connection Complete event). At connection setup, only the device(s) with the Authentication_Enable parameter enabled will try to authenticate the other device.

Note: Changing this parameter does not affect existing connections.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Authentication_Enable command succeeded.
0x01-0xFF	Read_Authentication_Enable command failed.See Table 6.1 on page 745 for list of Error Codes.

Authentication_Enable: *Size: 1 Byte*

Value	Parameter Description
0x00	Authentication disabled.
0x01	Authentication enabled for all connections.
0x02-0xFF	Reserved

Event(s) generated (unless masked away):

When the Read_Authentication_Enable command has completed, a Command Complete event will be generated.

4.7.24 Write_Authentication_Enable

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

Description:

This command will write the value for the Authentication_Enable parameter. The Authentication_Enable parameter controls if the local device requires to authenticate the remote device at connection setup (between the Create_Connection command or acceptance of an incoming ACL connection and the corresponding Connection Complete event). At connection setup, only the device(s) with the Authentication_Enable parameter enabled will try to authenticate the other device.

Note: Changing this parameter does not affect existing connections.

Command Parameters:

Authentication_Enable: *Size: 1 Byte*

Value	Parameter Description
0x00	Authentication disabled. **Default.**
0x01	Authentication enabled for all connection.
0x02-0xFF	Reserved

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write Authentication_Enable command succeeded.
0x01-0xFF	Write Authentication_Enable command failed. See Table 6.1 on page 745 for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Authentication_Enable command has completed, a Command Complete event will be generated.

Bluetooth

4.7.25 Read_Encryption_Mode

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Encryption_Mode	0x0021		Status, Encryption_Mode

Description:

This command will read the value for the Encryption_Mode parameter. The Encryption_Mode parameter controls if the local device requires encryption to the remote device at connection setup (between the Create_Connection command or acceptance of an incoming ACL connection and the corresponding Connection Complete event). At connection setup, only the device(s) with the Authentication_Enable parameter enabled and Encryption_Mode parameter enabled will try to encrypt the connection to the other device.
Note: Changing this parameter does not affect existing connections.

Command Parameters:

None.

Return Parameters:

Status: Size: 1 Byte

Value	Parameter Description
0x00	Read_Encryption_Mode command succeeded.
0x01-0xFF	Read_Encryption_Mode command failed. See Table 6.1 on page 745 for list of Error Codes.

Encryption_Mode: Size: 1 Byte

Value	Parameter Description
0x00	Encryption disabled.
0x01	Encryption only for point-to-point packets.
0x02	Encryption for both point-to-point and broadcast packets.
0x03-0xFF	Reserved.

Event(s) generated (unless masked away):

When the Read_Encryption_Mode command has completed, a Command Complete event will be generated.

4.7.26 Write_Encryption_Mode

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Encryption_Mode	0x0022	Encryption_Mode	Status

Description:

This command will write the value for the Encryption_Mode parameter. The Encryption_Mode parameter controls if the local device requires encryption to the remote device at connection setup (between the Create_Connection command or acceptance of an incoming ACL connection and the corresponding Connection Complete event). At connection setup, only the device(s) with the Authentication_Enable parameter enabled and Encryption_Mode parameter enabled will try to encrypt the connection to the other device.
Note: Changing this parameter does not affect existing connections.
A temporary link key must be used when both broadcast and point-to-point traffic shall be encrypted.

The Host must not specify the Encryption_Mode parameter with more encryption capability than its local device currently supports, although the parameter is used to request the encryption capability to the remote device. Note that the Host must not request the command with the Encryption_Mode parameter set to either 0x01 or 0x02, when the local device does not support encryption. Also note that the Host must not request the command with the parameter set to 0x02, when the local device does not support broadcast encryption.

Note that the actual Encryption_Mode to be returned in an event for a new connection (or in a Connection Complete event) will only support a part of the capability, when the local device requests more encryption capability than the current remote device supports. For example, 0x00 will always be returned in the event when the remote device supports no encryption, and either 0x00 or 0x01 will be returned when it supports only point-to-point encryption.

Command Parameters:

Encryption_Mode: *Size: 1 Byte*

Value	Parameter Description
0x00	Encryption disabled. **Default.**
0x01	Encryption only for point-to-point packets.
0x02	Encryption for both point-to-point and broadcast packets.
0x03-0xFF	Reserved.

Bluetooth

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write_Encryption_Mode command succeeded.
0x01-0xFF	Write_Encryption_Mode command failed. See Table 6.1 on page 745 for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Encryption_Mode command has completed, a Command Complete event will be generated.

Bluetooth

4.7.27 Read_Class_of_Device

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Class_of_Device	0x0023		Status, Class_of_Device

Description:

This command will read the value for the Class_of_Device parameter. The Class_of_Device parameter is used to indicate the capabilities of the local device to other devices.

Command Parameters:

None.

Return Parameters:

Status: Size: 1 Byte

Value	Parameter Description
0x00	Read_Class_of_Device command succeeded.
0x01-0xFF	Read_Class_of_Device command failed. See Table 6.1 on page 745 for list of Error Codes.

Class_of_Device: Size: 3 Bytes

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

Event(s) generated (unless masked away):

When the Read_Class_of_Device command has completed, a Command Complete event will be generated.

Bluetooth

4.7.28 Write_Class_of_Device

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Class_of_Device	0x0024	Class_of_Device	Status

Description:

This command will write the value for the Class_of_Device parameter. The Class_of_Device parameter is used to indicate the capabilities of the local device to other devices.

Command Parameters:

Class_of_Device: *Size: 3 Bytes*

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

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write_Class_of_Device command succeeded.
0x01-0xFF	Write_Class_of_Device command failed. See Table 6.1 on page 745 for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Class_of_Device command has completed, a Command Complete event will be generated.

4.7.29 Read_Voice_Setting

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

Description:

This command will read the values for the Voice_Setting parameter. The Voice_Setting parameter controls all the various settings for voice connections. These settings apply to all voice connections, and **cannot** be set for individual voice connections. The Voice_Setting parameter controls the configuration for voice connections: Input Coding, Air coding format, input data format, Input sample size, and linear PCM parameter.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Voice_Setting command succeeded.
0x01-0xFF	Read_Voice_Setting command failed. See Table 6.1 on page 745 for list of Error Codes.

Voice_Setting: *Size: 2 Bytes (10 Bits meaningful)*

Value	Parameter Description
00XXXXXXXX	Input Coding: Linear
01XXXXXXXX	Input Coding: μ-law Input Coding
10XXXXXXXX	Input Coding: A-law Input Coding
11XXXXXXXX	Reserved for Future Use
XX00XXXXXX	Input Data Format: 1's complement
XX01XXXXXX	Input Data Format: 2's complement
XX10XXXXXX	Input Data Format: Sign-Magnitude
XX11XXXXXX	Reserved for Future Use
XXXX0XXXXX	Input Sample Size: 8-bit (only for Liner PCM)
XXXX1XXXXX	Input Sample Size: 16-bit (only for Liner PCM)

Bluetooth

Value	Parameter Description
XXXXXnnnXX	Linear_PCM_Bit_Pos: # bit positions that MSB of sample is away from starting at MSB (only for Liner PCM).
XXXXXXXX00	Air Coding Format: CVSD
XXXXXXXX01	Air Coding Format: μ-law
XXXXXXXX10	Air Coding Format: A-law
XXXXXXXX11	Reserved

Event(s) generated (unless masked away):

When the Read_Voice_Setting command has completed, a Command Complete event will be generated.

Bluetooth

4.7.30 Write_Voice_Setting

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Voice_Setting	0x0026	Voice_Setting	Status

Description:

This command will write the values for the Voice_Setting parameter. The Voice_Setting parameter controls all the various settings for the voice connections. These settings apply to all voice connections and **cannot** be set for individual voice connections. The Voice_Setting parameter controls the configuration for voice connections: Input Coding, Air coding format, input data format, Input sample size, and linear PCM parameter.

Command Parameters:

Voice_Setting: *Size: 2 Bytes (10 Bits meaningful)*

Value	Parameter Description
00XXXXXXXX	Input Coding: Linear
01XXXXXXXX	Input Coding: μ-law Input Coding
10XXXXXXXX	Input Coding: A-law Input Coding
11XXXXXXXX	Reserved for Future Use
XX00XXXXXX	Input Data Format: 1's complement
XX01XXXXXX	Input Data Format: 2's complement
XX10XXXXXX	Input Data Format: Sign-Magnitude
XX11XXXXXX	Reserved for Future Use
XXXX0XXXXX	Input Sample Size: 8 bit (only for Liner PCM)
XXXX1XXXXX	Input Sample Size: 16 bit (only for Liner PCM)
XXXXXnnnXX	Linear_PCM_Bit_Pos: # bit positions that MSB of sample is away from starting at MSB (only for Liner PCM)
XXXXXXXX00	Air Coding Format: CVSD
XXXXXXXX01	Air Coding Format: μ-law
XXXXXXXX10	Air Coding Format: A-law
XXXXXXXX11	Reserved
0001100000	Default Condition

Bluetooth

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write_Voice_Setting command succeeded.
0x01-0xFF	Write_Voice_Setting command failed. See Table 6.1 on page 745 for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Voice_Setting command has completed, a Command Complete event will be generated.

4.7.31 Read_Automatic_Flush_Timeout

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Automatic_Flush_ Timeout	0x0027	Connection_Handle	Status, Connection_Handle, Flush_Timeout

Description:

This command will read the value for the Flush_Timeout parameter for the specified connection handle. The Flush_Timeout parameter is used for ACL connections ONLY. The Flush_Timeout parameter defines the amount of time before all chunks of the L2CAP packet, of which a baseband packet is currently being transmitted, are automatically flushed by the Host Controller. The timeout period starts when a transmission attempt is made for the first baseband packet of an L2CAP packet. This allows ACL packets to be automatically flushed without the Host device issuing a Flush command. The Read_Automatic_Flush_Timeout command provides support for isochronous data, such as video. When the L2CAP packet that is currently being transmitted is automatically 'flushed', the Failed Contact Counter is incremented by one. The first chunk of the next L2CAP packet to be transmitted for the specified connection handle may already be stored in the Host Controller. In that case, the transmission of the first baseband packet containing data from that L2CAP packet can begin immediately. If the next L2CAP packet is not stored in the Host Controller, all data that is sent to the Host Controller after the flush for the same connection handle will be discarded by the Host Controller until an HCI Data Packet having the start Packet_Boundary_Flag (0x02) is received. When this happens, a new transmission attempt will be made.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's Flush Timeout to read. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Automatic_Flush_Timeout command succeeded.
0x01-0xFF	Read_Automatic_Flush_Timeout command failed. See Table 6.1 on page 745 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's Flush Timeout has been read. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Flush_Timeout: *Size: 2 Bytes*

Value	Parameter Description
0	Timeout = ∞; No Automatic Flush
N = 0xXXXX	Flush Timeout = N * 0.625 msec Size: 11 bits Range: 0x0001 – 0x07FF

Event(s) generated (unless masked away):

When the Read_Automatic_Flush_Timeout command has completed, a Command Complete event will be generated.

4.7.32 Write_Automatic_Flush_Timeout

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Automatic_Flush_ Timeout	0x0028	Connection_Handle, Flush_Timeout	Status, Connection_Handle

Description:

This command will write the value for the Flush_Timeout parameter for the specified connection handle. The Flush_Timeout parameter is used for ACL connections ONLY. The Flush_Timeout parameter defines the amount of time before all chunks of the L2CAP packet, of which a baseband packet is currently being transmitted, are automatically flushed by the Host Controller. The timeout period starts when a transmission attempt is made for the first baseband packet of an L2CAP packet. This allows ACL packets to be automatically flushed without the Host device issuing a Flush command. The Write_Automatic_Flush_Timeout command provides support for isochronous data, such as video. When the L2CAP packet that is currently being transmitted is automatically 'flushed', the Failed Contact Counter is incremented by one. The first chunk of the next L2CAP packet to be transmitted for the specified connection handle may already be stored in the Host Controller. In that case, the transmission of the first baseband packet containing data from that L2CAP packet can begin immediately. If the next L2CAP packet is not stored in the Host Controller, all data that is sent to the Host Controller after the flush for the same connection handle will be discarded by the Host Controller until an HCI Data Packet having the start Packet_Boundary_Flag (0x02) is received. When this happens, a new transmission attempt will be made.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's Flush Timeout to write to. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Flush_Timeout: *Size: 2 Bytes*

Value	Parameter Description
0	Timeout = ∞; No Automatic Flush. **Default.**
N = 0xXXXX	Flush Timeout = N * 0.625 msec Size: 11 bits Range: 0x0001 – 0x07FF

Bluetooth

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write_Automatic_Flush_Timeout command succeeded.
0x01-0xFF	Write_Automatic_Flush_Timeout command failed. See Table 6.1 on page 745 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's Flush Timeout has been written. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Event(s) generated (unless masked away):

When the Write_Automatic_Flush_Timeout command has completed, a Command Complete event will be generated.

Bluetooth

4.7.33 Read_Num_Broadcast_Retransmissions

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Num_Broadcast_ Retransmissions	0x0029		Status, Num_Broadcast_Retran

Description:

This command will read the device's parameter value for the Number of Broad-cast Retransmissions. Broadcast packets are not acknowledged and are unre-liable. The Number of Broadcast Retransmissions parameter is used to increase the reliability of a broadcast message by retransmitting the broadcast message multiple times. This parameter defines the number of times the device will retransmit a broadcast data packet. This parameter should be adjusted as the link quality measurement changes.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Num_Broadcast_Retransmissions command succeeded.
0x01-0xFF	Read_Num_Broadcast_Retransmissions command failed. See Table 6.1 on page 745 for list of Error Codes.

Num_Broadcast_Retran: *Size: 1 Byte*

Value	Parameter Description
N = 0xXX	Number of Broadcast Retransmissions = N Range 0x00-0xFF

Event(s) generated (unless masked away):

When the Read_Num_Broadcast_Retransmission command has completed, a Command Complete event will be generated.

Bluetooth

4.7.34 Write_Num_Broadcast_Retransmissions

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Num_Broadcast_ Retransmissions	0x002A	Num_Broadcast_Retran	Status

Description:

This command will write the device's parameter value for the Number of Broadcast Retransmissions. Broadcast packets are not acknowledged and are unreliable. The Number of Broadcast Retransmissions parameter is used to increase the reliability of a broadcast message by retransmitting the broadcast message multiple times. This parameter defines the number of times the device will retransmit a broadcast data packet. This parameter should be adjusted as link quality measurement change.

Command Parameters:

Num_Broadcast_Retran: *Size: 1 Byte*

Value	Parameter Description
N = 0xXX	Number of Broadcast Retransmissions = N Range 0x00-0xFF Default: N = 0x01

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write_Num_Broadcast_Retransmissions command succeeded.
0x01-0xFF	Write_Num_Broadcast_Retransmissions command failed. See Table 6.1 on page 745 for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Num_Broadcast_Retransmissions command has completed, a Command Complete event will be generated.

4.7.35 Read_Hold_Mode_Activity

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Hold_Mode_Activity	0x002B		Status, Hold_Mode_Activity

Description:

This command will read the value for the Hold_Mode_Activity parameter. The Hold_Mode_Activity value is used to determine what activities should be suspended when the device is in hold mode. After the hold period has expired, the device will return to the previous mode of operation. Multiple hold mode activities may be specified for the Hold_Mode_Activity parameter by performing a bitwise OR operation of the different activity types. If no activities are suspended, then all of the current Periodic Inquiry, Inquiry Scan, and Page Scan settings remain valid during the Hold Mode. If the Hold_Mode_Activity parameter is set to Suspend Page Scan, Suspend Inquiry Scan, and Suspend Periodic Inquiries, then the device can enter a low-power state during the Hold Mode period, and all activities are suspended. Suspending multiple activities can be specified for the Hold_Mode_Activity parameter by performing a bitwise OR operation of the different activity types.The Hold Mode Activity is only valid if all connections are in Hold Mode.

Command Parameters:

None.

Return Parameters:

Status: Size: 1 Byte

Value	Parameter Description
0x00	Read_Hold_Mode_Activity command succeeded.
0x01-0xFF	Read_Hold_Mode_Activity command failed. See Table 6.1 on page 745 for list of Error Codes.

Hold_Mode_Activity: Size: 1 Byte

Value	Parameter Description
0x00	Maintain current Power State.
0x01	Suspend Page Scan.
0x02	Suspend Inquiry Scan.
0x04	Suspend Periodic Inquiries.
0x08-0xFF	Reserved for Future Use.

Event(s) generated (unless masked away):

When the Read_Hold_Mode_Activity command has completed, a Command Complete event will be generated.

Host Controller Interface Functional Specification

Bluetooth

4.7.36 Write_Hold_Mode_Activity

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Hold_Mode_Activity	0x002C	Hold_Mode_Activity	Status

Description:

This command will write the value for the Hold_Mode_Activity parameter. The Hold_Mode_Activity value is used to determine what activities should be suspended when the device is in hold mode. After the hold period has expired, the device will return to the previous mode of operation. Multiple hold mode activities may be specified for the Hold_Mode_Activity parameter by performing a bitwise OR operation of the different activity types. If no activities are suspended, then all of the current Periodic Inquiry, Inquiry Scan, and Page Scan settings remain valid during the Hold Mode. If the Hold_Mode_Activity parameter is set to Suspend Page Scan, Suspend Inquiry Scan, and Suspend Periodic Inquiries, then the device can enter a low power state during the Hold Mode period and all activities are suspended. Suspending multiple activities can be specified for the Hold_Mode_Activity parameter by performing a bitwise OR operation of the different activity types. The Hold Mode Activity is only valid if all connections are in Hold Mode.

Command Parameters:

Hold_Mode_Activity: *Size: 1 Byte*

Value	Parameter Description
0x00	Maintain current Power State. **Default.**
0x01	Suspend Page Scan.
0x02	Suspend Inquiry Scan.
0x04	Suspend Periodic Inquiries.
0x08-0xFF	Reserved for Future Use.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write_Hold_Mode_Activity command succeeded.
0x01-0xFF	Write_Hold_Mode_Activity command failed. See Table 6.1 on page 745 for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Hold_Mode_Activity command has completed, a Command Complete event will be generated.

Bluetooth

4.7.37 Read_Transmit_Power_Level

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Transmit_ Power_Level	0x002D	Connection_Handle, Type	Status, Connection_Handle, Transmit_Power_Level

Description:

This command will read the values for the Transmit_Power_Level parameter for the specified Connection Handle. The Connection_Handle must be a Connection_Handle for an ACL connection.

Command Parameters:

_Connection_Handle:_   _Size: 2 Bytes (12 Bits meaningful)_

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's Transmit Power Level setting to read. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Type: _Size: 1 Byte_

Value	Parameter Description
0x00	Read Current Transmit Power Level.
0x01	Read Maximum Transmit Power Level.
0x02-0xFF	Reserved

Return Parameters:

Status: _Size: 1 Byte_

Value	Parameter Description
0x00	Read_Transmit_Power_Level command succeeded.
0x01-0xFF	Read_Transmit_Power_Level command failed. See Table 6.1 on page 745 for list of Error Codes.

_Connection_Handle:_   _Size: 2 Bytes (12 Bits meaningful)_

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's Transmit Power Level setting is returned. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Bluetooth

Transmit_Power_Level: *Size: 1 Byte*

Value	Parameter Description
N = 0xXX	Size: 1 Byte (signed integer) Range: -30 $\leq$ N $\leq$ 20 Units: dBm

Event(s) generated (unless masked away):

When the Read_Transmit_Power_Level command has completed, a Command Complete event will be generated.

Bluetooth

4.7.38 Read_SCO_Flow_Control_Enable

Command	OCF	Command Parameters	Return Parameters
HCI_Read_SCO_ Flow_Control_Enable	0x002E		Status, SCO_Flow_Control_Enable

Description:

The Read_SCO_Flow_Control_Enable command provides the ability to read
the SCO_Flow_Control_Enable setting. By using this setting, the Host can
decide if the Host Controller will send Number Of Completed Packets events
for SCO Connection Handles. This setting allows the Host to enable and dis-
able SCO flow control.
Note: the SCO_Flow_Control_Enable setting can only be changed if no con-
nections exist.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_SCO_Flow_Control_Enable command succeeded
0x01-0xFF	Read_SCO_Flow_Control_Enable command failed see Table 6.1 on page 746 for list of Error Codes

SCO_Flow_Control_Enable: *Size: 1 Byte*

Value	Parameter Description
0x00	SCO Flow Control is disabled. No Number of Completed Packets events will be sent from the Host Controller for SCO Connection Handles.
0x01	SCO Flow Control is enabled. Number of Completed Packets events will be sent from the Host Controller for SCO Connection Handles.

Event(s) generated (unless masked away):

When the Read_SCO_Flow_Control_Enable command has completed a Com-
mand Complete event will be generated.

4.7.39 Write_SCO_Flow_Control_Enable

Command	OCF	Command Parameters	Return Parameters
HCI_Write_SCO_Flow _Control_Enable	0x002F	SCO_Flow_Control_Enable	Status

Description:

The Write_SCO_Flow_Control_Enable command provides the ability to write the SCO_Flow_Control_Enable setting. By using this setting, the Host can decide if the Host Controller will send Number Of Completed Packets events for SCO Connection Handles. This setting allows the Host to enable and disable SCO flow control.
Note: the SCO_Flow_Control_Enable setting can only be changed if no connections exist.

Command Parameters:

SCO_Flow_Control_Enable: *Size: 1 Byte*

Value	Parameter Description
0x00	SCO Flow Control is disabled. No Number of Completed Packets events will be sent from the Host Controller for SCO Connection Handles. **Default**
0x01	SCO Flow Control is enabled. Number of Completed Packets events will be sent from the Host Controller for SCO Connection Handles.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write_SCO_Flow_Control_Enable command succeeded
0x01-0xFF	Write_SCO_Flow_Control_Enable command failed see Table 6.1 on page 746 for list of Error Codes

Event(s) generated (unless masked away):

When the Write_SCO_Flow_Control_Enable command has completed a Command Complete event will be generated.

Bluetooth

4.7.40 Set_Host_Controller_To_Host_Flow_Control

Command	OCF	Command Parameters	Return Parameters
HCI_Set_Host_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 in the direction from the Host Controller to the Host. If flow control is turned off, the Host should not send the Host_Number_Of_Completed_Packets command. That command will be ignored by the Host Controller if it is sent by the Host and flow control is off.

Command Parameters:

Flow_Control_Enable: *Size: 1 Byte*

Value	Parameter Description
0x00	Flow control **off** in direction from Host Controller to Host. **Default.**
0x01	Flow control **on** in direction from Host Controller to Host.
0x02-0xFF	Reserved

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Set_Host_Controller_To_Host_Flow_Control command succeeded.
0x01-0xFF	Set_Host_Controller_To_Host_Flow_Control command failed. See Table 6.1 on page 7450 for list of Error Codes.

Event(s) generated (unless masked away):

When the Set_Host_Controller_To_Host_Flow_Control command has completed, a Command Complete event will be generated.

Bluetooth.

4.7.41 Host_Buffer_Size

Command	OCF	Command Parameters	Return Parameters
HCI_Host_Buffer_Size	0x0033	Host_ACL_Data_Packet_Length, Host_SCO_Data_Packet_Length, Host_Total_Num_ACL_Data_Packets, Host_Total_Num_SCO_Data_Packets	Status

Description:

The Host_Buffer_Size command is used by the Host to notify the Host Controller about the maximum size of the data portion of HCI ACL and SCO Data Packets sent from the Host Controller to the Host. The Host Controller will segment the data to be transmitted from the Host Controller to the Host according to these sizes, so that the HCI Data Packets will contain data with up to these sizes. The Host_Buffer_Size command also notifies the Host Controller about the total number of HCI ACL and SCO Data Packets that can be stored in the data buffers of the Host. If flow control from the Host Controller to the Host is turned off, and the Host_Buffer_Size command has not been issued by the Host, this means that the Host Controller will send HCI Data Packets to the Host with any lengths the Host Controller wants to use, and it is assumed that the data buffer sizes of the Host are unlimited. If flow control from the Host controller to the Host is turned on, the Host_Buffer_Size command must after a power-on or a reset always be sent by the Host before the first Host_Number_Of_Completed_Packets command is sent.

(The Set_Host_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 Host Controller to the Host. The Host_SCO_Data_Packet_Length command parameter is used to determine the maximum size of HCI SCO Data Packets. Both the Host and the Host Controller must support command and event packets, where the data portion (excluding header) contained in the packets is 255 bytes 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 Host Controller will determine how the buffers are to be divided between different Connection Handles. The Host_Total_Num_SCO_Data_ Packets command parameter gives the same information for HCI SCO Data Packets.
Note: the Host_ACL_Data_Packet_Length and Host_SCO_Data_Packet_ Length command parameters do not include the length of the HCI Data Packet header.

Bluetooth.

Command Parameters:

Host_ACL_Data_Packet_Length: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	Maximum length (in bytes) of the data portion of each HCI ACL Data Packet that the Host is able to accept.

Host_SCO_Data_Packet_Length: *Size: 1 Byte*

Value	Parameter Description
0xXX	Maximum length (in bytes) of the data portion of each HCI SCO Data Packet that the Host is able to accept.

Host_Total_Num_ACL_Data_Packets: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	Total number of HCI ACL Data Packets that can be stored in the data buffers of the Host.

Host_Total_Num_SCO_Data_Packets: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	Total number of HCI SCO Data Packets that can be stored in the data buffers of the Host.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Host_Buffer_Size command succeeded.
0x01-0xFF	Host_Buffer_Size command failed. See Table 6.1 on page 745 for list of Error Codes.

Event(s) generated (unless masked away):

When the Host_Buffer_Size command has completed, a Command Complete event will be generated.

Bluetooth

4.7.42 Host_Number_Of_Completed_Packets

Command	OCF	Command Parameters	Return Parameters
HCI_Host_Number_Of_ Completed_Packets	0x0035	Number_Of_Handles, Connection_ Handle[i], Host_Num_Of_Completed_Packets [i]	

Description:

The Host_Number_Of_Completed_Packets command is used by the Host to indicate to the Host Controller the number of HCI Data Packets that have been completed for each Connection Handle since the previous Host_Number_Of_ Completed_Buffers command was sent to the Host Controller. This means that the corresponding buffer space has been freed in the Host. Based on this information, and the Host_Total_Num_ACL_Data_Packets and Host_Total_Num_ SCO_Data_Packets command parameters of the Host_Buffer_Size command, the Host Controller can determine for which Connection Handles the following HCI Data Packets should be sent to the Host. The command should only be issued by the Host if flow control in the direction from the Host Controller to the Host is on and there is at least one connection, or if the Host Controller is in local loopback mode. Otherwise, the command will be ignored by the Host Controller. While the Host has HCI Data Packets in its buffers, it must keep sending the Host_Number_Of_Completed_Packets command to the Host Controller at least periodically, until it finally reports that all buffer space in the Host used by ACL Data Packets has been freed. The rate with which this command is sent is manufacturer specific.

(The Set_Host_Controller_To_Host_Flow_Control command is used to turn flow control on or off.) If flow control from the Host controller to the Host is turned on, the Host_Buffer_Size command must after a power-on or a reset always be sent by the Host before the first Host_Number_Of_Completed_Packets command is sent.
Note: the Host_Number_Of_Completed_Packets command is a special command in the sense that no event is normally generated after the command has completed. The command may be sent at any time by the Host when there is at least one connection, or if the Host Controller is in local loopback mode independent of other commands. The normal flow control for commands is not used for the Host_Number_Of_Completed_Packets command.

Bluetooth

Command Parameters:

Number_Of_Handles: *Size: 1 Byte*

Value	Parameter Description
0xXX	The number of Connection Handles and Host_Num_Of_Completed_Packets parameters pairs contained in this command. Range: 0-255

Connection_Handle[i]: Size: Number_Of_Handles*2 Bytes (12 Bits meaningful)

Value	Parameter Description
0xXXXX	Connection Handle Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Host_Num_Of_Completed_Packets [i]:   Size: Number_Of_Handles * 2 Bytes

Value	Parameter Description
N = 0xXXXX	The number of HCI Data Packets that have been completed for the associated Connection Handle since the previous time the event was returned. Range for N: 0x0000-0xFFFF

Return Parameters:

None.

Event(s) generated (unless masked away):

Normally, no event is generated after the Host_Number_Of_Completed_ Packets command has completed. However, if the Host_Number_Of_ Completed_Packets command contains one or more invalid parameters, the Host Controller will return a Command Complete event with a failure status indicating the Invalid HCI Command Parameters error code. The Host may send the Host_Number_Of_Completed_Packets command at any time when there is at least one connection, or if the Host Controller is in local loopback mode. The normal flow control for commands is not used for this command.

Bluetooth

4.7.43 Read_Link_Supervision_Timeout

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Link_Supervision_Timeout	0x0036	Connection_Handle	Status, Connection_Handle, Link_Supervision_Timeout

Description:

This command will read the value for the Link_Supervision_Timeout parameter for the device. The Link_Supervision_Timeout parameter is used by the master or slave Bluetooth device to monitor link loss. If, for any reason, no Baseband packets are received from that Connection Handle for a duration longer than the Link_Supervision_Timeout, the connection is disconnected. The same timeout value is used for both SCO and ACL connections for the device specified by the Connection Handle.
Note: the Connection_Handle used for this command must be the ACL connection to the appropriate device. This command will set the Link_Supervision_Timeout values for other SCO Connection_Handle to that device.

Note: Setting the Link_Supervision_Timeout to No Link_Supervision_Timeout (0x0000) will disable the Link_Supervision_Timeout check for the specified Connection Handle. This makes it unnecessary for the master of the piconet to unpark and then park each Bluetooth Device every ~40 seconds. By using the No Link_Supervision_Timeout setting, the scalability of the Park mode is not limited.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's Link Supervision Timeout value is to be read. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Link_Supervision_Timeout command succeeded.
0x01-0xFF	Read_Link_Supervision_Timeout command failed. See Table 6.1 on page 745 for list of Error Codes.

Bluetooth

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's Link Supervision Timeout value was read.
	Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Link_Supervision_Timeout: *Size: 2 Bytes*

Value	Parameter Description
0x0000	No Link_Supervision_Timeout.
N = 0xXXXX	Measured in Number of Baseband slots
	Link_Supervision_Timeout = N * 0.625 msec (1 Baseband slot)
	Range for N: 0x0001 – 0xFFFF
	Time Range: 0.625ms - 40.9 sec

Event(s) generated (unless masked away):

When the Read_Link_Supervision_Timeout command has completed, a Command Complete event will be generated.

Bluetooth

4.7.44 Write Link_Supervision_Timeout

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Link_Supervision_Timeout	0x0037	Connection_Handle, Link_Supervision_Timeout	Status, Connection_Handle

Description:

This command will write the value for the Link_Supervision_Timeout parameter for the device. The Link_Supervision_Timeout parameter is used by the master or slave Bluetooth device to monitor link loss. If, for any reason, no Baseband packets are received from that Connection_Handle for a duration longer than the Link_Supervision_Timeout, the connection is disconnected. The same timeout value is used for both SCO and ACL connections for the device specified by the Connection_Handle.

Note: the Connection_Handle used for this command must be the ACL connection to the appropriate device. This command will set the Link_Supervision_Timeout values for other SCO Connection_Handle to that device.

Note: Setting the Link_Supervision_Timeout parameter to No Link_Supervision_Timeout (0x0000) will disable the Link_Supervision_Timeout check for the specified Connection Handle. This makes it unnecessary for the master of the piconet to unpark and then park each Bluetooth Device every ~40 seconds. By using the No Link_Supervision_Timeout setting, the scalability of the Park mode is not limited.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's Link Supervision Timeout value is to be written. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Link_Supervision_Timeout: *Size: 2 Bytes*

Value	Parameter Description
0x0000	No Link_Supervision_Timeout.
N = 0xXXXX	Measured in Number of Baseband slots Link_Supervision_Timeout = N*0.625 msec (1 Baseband slot) Range for N: 0x0001 – 0xFFFF Time Range: 0.625ms – 40.9 sec **Default**: N = 0x7D00 Link_Supervision_Timeout = 20 sec

Return Parameters:

Status: _Size: 1 Byte_

Value	Parameter Description
0x00	Write_Link_Supervision_Timeout command succeeded.
0x01-0xFF	Write_Link_Supervision_Timeout command failed. See Table 6.1 on page 745 for list of Error Codes.

_Connection_Handle:_   _Size: 2 Bytes (12 Bits meaningful)_

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's Link Supervision Timeout value was written.
	Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Event(s) generated (unless masked away):

When the Write_Link_Supervision_Timeout command has completed, a Command Complete event will be generated.

Bluetooth

4.7.45 Read_Number_Of_Supported_IAC

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Number_Of_Supported_IAC	0x0038		Status, Num_Support_IAC

Description:

This command will read the value for the number of Inquiry Access Codes (IAC) that the local Bluetooth device can simultaneous listen for during an Inquiry Scan. All Bluetooth devices are required to support at least one IAC, the General Inquiry Access Code (GIAC or UIAC), but some Bluetooth devices support additional IACs.

Command Parameters:

None

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Number_Of_Supported_IAC command succeeded.
0x01-0xFF	Read_Number_Of_Supported_IAC command failed. See Table 6.1 on page 745 for list of Error Codes.

Num_Support_IAC   *Size: 1 Byte*

Value	Parameter Description
0xXX	Specifies the number of Supported IAC that the local Bluetooth device can simultaneous listen for during an Inquiry Scan. Range: 0x01-0x40

Event(s) generated (unless masked away):

When the Read_Number_Of_Supported_IAC command has completed, a Command Complete event will be generated.

Bluetooth

4.7.46 Read_Current_IAC_LAP

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 Bluetooth device is simultaneously scanning for during Inquiry Scans. All Bluetooth devices are required to support at least one IAC (GIAC or UIAC). Some Bluetooth devices support additional IACs.

Command Parameters:

None

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Current_IAC_LAP command succeeded.
0x01-0xFF	Read_Current_IAC_LAP command failed. See Table 6.1 on page 745 for list of Error Codes.

Num_Current_IAC   *Size: 1 Byte*

Value	Parameter Description
0xXX	Specifies the number of IACs which are currently in use by the local Bluetooth device to simultaneously listen for during an Inquiry Scan. Range: 0x01-0x40

IAC_LAP[i]   *Size: 3 Bytes * Num_Current_IAC*

Value	Parameter Description
0xXXXXXX	LAPs used to create the IAC which is currently in use by the local Bluetooth device to simultaneously listen for during an Inquiry Scan. Range: 0x9E8B00-0x9E8B3F

Event(s) generated (unless masked away):

When the Read_Current_IAC_LAP command has completed, a Command Complete event will be generated.

Bluetooth

4.7.47 Write_Current_IAC_LAP

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 Bluetooth device is simultaneously scanning for during Inquiry Scans. All Bluetooth devices are required to support at least one IAC (GIAC or UIAC). Some Bluetooth devices support additional IACs. Therefore, the LAP used to create the GIAC or UIAC must be among the IAC_LAP parameters of this command.
Note: this command writes over the current IACs used by the Bluetooth device. If the value of the NumCurrentIAC is more than the number of supported IACs, then only the first, X Inquiry Access Codes (where X equals the number of supported IACs) will be stored without any error.

Command Parameters:

Num_Current_IAC   *Size: 1 Byte*

Value	Parameter Description
0xXX	Specifies the number of IACs which are currently in use by the local Bluetooth device to simultaneously listen for during an Inquiry Scan. Range: 0x01-0x40

IAC_LAP[i]   *Size: 3 Bytes * Num_Current_IAC*

Value	Parameter Description
0xXXXXXX	LAP(s) used to create IAC which is currently in use by the local Bluetooth device to simultaneously listen for during an Inquiry Scan. Range: 0x9E8B00-0x9E8B3F. The GIAC is the default IAC to be used. If additional IACs are supported, additional default IAC will be determined by the manufacturer.

Bluetooth

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write_Current_IAC_LAP command succeeded.
0x01-0xFF	Write_Current_IAC_LAP command failed. See Table 6.1 on page 745 for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Current_IAC_LAP command has completed, a Command Complete event will be generated.

Bluetooth

4.7.48 Read_Page_Scan_Period_Mode

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Page_Scan_Period_ Mode	0x003B		Status, Page_Scan_Period_Mode

Description:

This command is used to read the mandatory Page_Scan_Period_Mode of the local Bluetooth device. Every time an inquiry response message is sent, the Bluetooth device will start a timer (T_mandatory_pscan), the value of which is dependent on the Page_Scan_Period_Mode. As long as this timer has not expired, the Bluetooth device will use the Page_Scan_Period_Mode for all following page scans.
Note: the timer T_mandatory_pscan will be reset at each new inquiry response. For details see the "Baseband Specification" on page 33. (Keyword: SP-Mode, FHS-Packet, T_mandatory_pscan, Inquiry-Response).

After transmitting one or more inquiry response (FHS) packets as a result of the inquiry scan process, the local Bluetooth device is allowed to enter the page scan state using mandatory page scan mode regardless of the setting of the Scan_Enable parameter.

Command Parameters:

None

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Page_Scan_Period_Mode command succeeded.
0x01-0xFF	Read_Page_Scan_Period_Mode command failed. See Table 6.1 on page 745 for list of Error Codes.

Page_Scan_Period_Mode: *Size: 1 Byte*

Value	Parameter Description
0x00	P0
0x01	P1
0x02	P2
0x03-0xFF	Reserved.

Bluetooth

Event(s) generated (unless masked away):

When the Read_Page_Scan_Period_Mode command has completed, a Command Complete event will be generated.

Bluetooth

4.7.49 Write_Page_Scan_Period_Mode

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Page_Scan_Period_Mode	0x003C	Page_Scan_Period_Mode	Status

Description:

This command is used to write the mandatory Page_Scan_Period_Mode of the local Bluetooth device. Every time an inquiry response message is sent, the Bluetooth device will start a timer (T_mandatory_pscan), the value of which is dependent on the Page_Scan_Period_Mode. As long as this timer has not expired, the Bluetooth device will use the Page_Scan_Period_Mode for all following page scans.
Note: the timer T_mandatory_pscan will be reset at each new inquiry response. For details see the "Baseband Specification" on page 33. (Keyword: SP-Mode, FHS-Packet, T_mandatory_pscan, Inquiry-Response).

After transmitting one or more inquiry response (FHS) packets as a result of the inquiry scan process, the local Bluetooth device is allowed to enter the page scan state using mandatory page scan mode regardless of the setting of the Scan_Enable parameter.

Command Parameters:

Page_Scan_Period_Mode: *Size: 1 Byte*

Value	Parameter Description
0x00	P0. **Default.**
0x01	P1
0x02	P2
0x03-0xFF	Reserved.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write_Page_Scan_Period_Mode command succeeded.
0x01-0xFF	Write_Page_Scan_Period_Mode command failed. See Table 6.1 on page 745 for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Page_Scan_Period_Mode command has completed, a Command Complete event will be generated.

4.7.50 Read_Page_Scan_Mode

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Page_Scan_Mode	0x003D		Status, Page_Scan_Mode

Description:

This command is used to read the default page scan mode of the local Bluetooth device. The Page_Scan_Mode parameter indicates the page scan mode that is used for default page scan. Currently one mandatory page scan mode and three optional page scan modes are defined. Following an inquiry response, if the Baseband timer T_mandatory_pscan has not expired, the mandatory page scan mode must be applied. For details see the "Baseband Specification" on page 33 (Keyword: Page-Scan-Mode, FHS-Packet, T_mandatory_pscan)

Command Parameters:

None

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Page_Scan_Mode command succeeded.
0x01-0xFF	Read_Page_Scan_Mode command failed. See Table 6.1 on page 745 for list of Error Codes.

Page_Scan_Mode: *Size: 1 Byte*

Value	Parameter Description
0x00	Mandatory Page Scan Mode
0x01	Optional Page Scan Mode I
0x02	Optional Page Scan Mode II
0x03	Optional Page Scan Mode III
0x04-0xFF	Reserved

Event(s) generated (unless masked away):

When the Read_Page_Scan_Mode command has completed, a Command Complete event will be generated.

4.7.51 Write_Page_Scan_Mode

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Page_Scan_Mode	0x003E	Page_Scan_Mode	Status

Description:

This command is used to write the default page scan mode of the local Bluetooth device. The Page_Scan_Mode parameter indicates the page scan mode that is used for the default page scan. Currently, one mandatory page scan mode and three optional page scan modes are defined. Following an inquiry response, if the Baseband timer T_mandatory_pscan has not expired, the mandatory page scan mode must be applied. For details see the "Baseband Specification" on page 33. (Keyword: Page-Scan-Mode, FHS-Packet, T_mandatory_pscan).

Command Parameters:

Page_Scan_Mode: *Size: 1 Byte*

Value	Parameter Description
0x00	Mandatory Page Scan Mode. **Default**.
0x01	Optional Page Scan Mode I
0x02	Optional Page Scan Mode II
0x03	Optional Page Scan Mode III
0x04-0xFF	Reserved.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write_Page_Scan_Mode command succeeded.
0x01-0xFF	Write_Page_Scan_Mode command failed. See Table 6.1 on page 745 for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Page_Scan_Mode command has completed, a Command Complete event will be generated.

Bluetooth

4.8   INFORMATIONAL PARAMETERS

The Informational Parameters are fixed by the manufacturer of the Bluetooth hardware. These parameters provide information about the Bluetooth device and the capabilities of the Host Controller, Link Manager, and Baseband. The host device cannot modify any of these parameters. For Informational Parameters Commands, the OGF is defined as 0x04

Command	Command Summary Description
Read_Local_Version_Information	The Read_Local_Version_Information command will read the values for the version information for the local Bluetooth device.
Read_Local_Supported_Features	The Read_Local_Supported_Features command requests a list of the supported features for the local device.
Read_Buffer_Size	The Read_Buffer_Size command returns the size of the HCI buffers. These buffers are used by the Host Controller to buffer data that is to be transmitted.
Read_Country_Code	The Read_Country_Code command will read the value for the Country Code status parameter. The Country Code defines which range of frequency band of the ISM 2.4 GHz band will be used by the device.
Read_BD_ADDR	The Read_BD_ADDR command will read the value for the BD_ADDR parameter. The BD_ADDR is a 48-bit unique identifier for a Bluetooth device.

Bluetooth

4.8.1 Read_Local_Version_Information

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Local_Version_ Information	0x0001		Status, HCI Version, HCI Revision, LMP Version, Manufacturer_Name, LMP Subversion

Description:

This command will read the values for the version information for the local Bluetooth device. The version information consists of two parameters: the version and revision parameters.

The version parameter defines the major hardware version of the Bluetooth hardware. The version parameter only changes when new versions of the Bluetooth hardware are produced for new Bluetooth SIG specifications. The version parameter is controlled by the SIG.

The revision parameter should be controlled by the manufacturer and should be changed as needed. The Manufacturer_Name parameter indicates the manufacturer of the local Bluetooth module as specified by the LMP definition of this parameter. The subversion parameter should be controlled by the manufacturer and should be changed as needed. The subversion parameter defines the various revisions that each version of the Bluetooth hardware will go through as design processes change and errors are fixed. This allows the software to determine what Bluetooth hardware is being used, and to work around various bugs in the hardware if necessary.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Local_Version_Information command succeeded.
0x01-0xFF	Read_Local_Version_Information command failed. See Table 6.1 on page 745 for list of Error Codes.

Bluetooth.

HCI_Version: *Size: 1 Byte*

Value	Parameter Description
0xXX	Version of the Current HCI in the Bluetooth hardware.
	0x00: Bluetooth HCI Specification 1.0
	0x01-0xFF: Reserved

HCI_Revision: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	Revision of the Current HCI in the Bluetooth hardware.

LMP_Version: *Size: 1 Byte*

Value	Parameter Description
0xXX	Version of the Current LMP in the Bluetooth Hardware, see Table 5.2 on page 231 in the Link Manager Protocol for assigned values (VersNr).

Manufacturer_Name: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	Manufacturer Name of the Bluetooth Hardware, see Table 5.2 on page 231 in the Link Manager Protocol for assigned values (CompId).

LMP_Subversion: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	Subversion of the Current LMP in the Bluetooth Hardware, see Table 5.2 on page 231 in the Link Manager Protocol for assigned values (SubVersNr).

Event(s) generated (unless masked away):

When the Read_Local_Version_Information command has completed, a Command Complete event will be generated.

Bluetooth

4.8.2 Read_Local_Supported_Features

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 device. This command will return a list of the LMP features. For details see "Link Manager Protocol" on page 185.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Local_Supported_Features command succeeded.
0x01-0xFF	Read_Local_Supported_Features command failed. See Table 6.1 on page 745 for list of Error Codes.

LMP_Features: *Size: 8 Bytes*

Value	Parameter Description
0xXXXXXXXX XXXXXXXX	Bit Mask List of LMP features. For details see "Link Manager Protocol" on page 185

Event(s) generated (unless masked away):

When the Read_Local_Supported_Features command has completed, a Command Complete event will be generated.

Bluetooth

4.8.3 Read_Buffer_Size

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Buffer_Size	0x0005		Status, HC_ACL_Data_Packet_Length, HC_SCO_Data_Packet_Length, HC_ Total_Num_ACL_Data_Packets, HC_Total_Num_SCO_Data_Packets

Description:

The Read_Buffer_Size command is used to read the maximum size of the data portion of HCI ACL and SCO Data Packets sent from the Host to the Host Controller. The Host will segment the data to be transmitted from the Host to the Host Controller according to these sizes, so that the HCI Data Packets will contain data with up to these sizes. The Read_Buffer_Size command also returns the total number of HCI ACL and SCO Data Packets that can be stored in the data buffers of the Host Controller. The Read_Buffer_Size command must be issued by the Host before it sends any data to the Host Controller.

The HC_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 Host Controller to be broken up into baseband packets by the Link Manager. The HC_SCO_Data_Packet_Length return parameter is used to determine the maximum size of HCI SCO Data Packets. Both the Host and the Host Controller must support command and event packets, where the data portion (excluding header) contained in the packets is 255 bytes in size. The HC_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 Host Controller. The Host will determine how the buffers are to be divided between different Connection Handles. The
HC_Total_Num_SCO_Data_
Packets return parameter gives the same information but for HCI SCO Data Packets.
Note: the HC_ACL_Data_Packet_Length and HC_SCO_Data_Packet_Length return parameters do not include the length of the HCI Data Packet header.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Buffer_Size command succeeded.
0x01-0xFF	Read_Buffer_Size command failed. See Table 6.1 on page 745 for list of Error Codes.

HC_ACL_Data_Packet_Length: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	Maximum length (in bytes) of the data portion of each HCI ACL Data Packet that the Host Controller is able to accept.

HC_SCO_Data_Packet_Length: *Size: 1 Byte*

Value	Parameter Description
0xXX	Maximum length (in bytes) of the data portion of each HCI SCO Data Packet that the Host Controller is able to accept.

HC_Total_Num_ACL_Data_Packets: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	Total number of HCI ACL Data Packets that can be stored in the data buffers of the Host Controller.

HC_Total_Num_SCO_Data_Packets: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	Total number of HCI SCO Data Packets that can be stored in the data buffers of the Host Controller.

Event(s) generated (unless masked away):

When the Read_Buffer_Size command has completed, a Command Complete event will be generated.

Bluetooth

4.8.4 Read_Country_Code

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Country_Code	0x0007		Status, Country_Code

Description:

This command will read the value for the Country_Code return parameter. The Country_Code defines which range of frequency band of the ISM 2.4 GHz band will be used by the device. Each country has local regulatory bodies regulating which ISM 2.4 GHz frequency ranges can be used.

Command Parameters:

None.

Return Parameters:

Status: Size: 1 Byte

Value	Parameter Description
0x00	Read_Country_Code command succeeded.
0x01-0xFF	Read_Country_Code command failed. See Table 6.1 on page 745 for list of Error Codes.

Country_Code: Size: 1 Byte

Value	Parameter Description
0x00	North America & Europe*
0x01	France
0x02	Spain
0x03	Japan
0x04-FF	Reserved for Future Use.

*. Except Spain and France

Event(s) generated (unless masked away):

When the Read_Country_Code command has completed, a Command Complete event will be generated.

Bluetooth

4.8.5 Read_BD_ADDR

Command	OCF	Command Parameters	Return Parameters
HCI_Read_BD_ADDR	0x0009		Status, BD_ADDR

Description:

This command will read the value for the BD_ADDR parameter.
The BD_ADDR is a 48-bit unique identifier for a Bluetooth device. See the "Baseband Specification" on page 33 for details of how BD_ADDR is used.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_BD_ADDR command succeeded.
0x01-0xFF	Read_BD_ADDR command failed. See Table 6.1 on page 745 for list of Error Codes.

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device

Event(s) generated (unless masked away):

When the Read_BD_ADDR command has completed, a Command Complete event will be generated.

Bluetooth

4.9   STATUS PARAMETERS

The Host Controller modifies all status parameters. These parameters provide information about the current state of the Host Controller, Link Manager, and Baseband. The host device cannot modify any of these parameters other than to reset certain specific parameters. For the Status and baseband, the OGF is defined as 0x05

Command	Command Summary Description
Read_Failed_Contact_Counter	The Read_Failed_Contact_Counter will read the value for the Failed_Contact_Counter parameter for a particular connection to another device. The Failed_Contact_Counter records the number of consecutive incidents in which either the slave or master didn't respond after the flush timeout had expired, and the L2CAP packet that was currently being transmitted was automatically 'flushed'.
Reset_Failed_Contact_Counter	The Reset_Failed_Contact_Counter will reset the value for the Failed_Contact_Counter parameter for a particular connection to another device. The Failed_Contact_Counter records the number of consecutive incidents in which either the slave or master didn't respond after the flush timeout had expired and the L2CAP packet that was currently being transmitted was automatically 'flushed'.
Get_Link_Quality	The Get_Link_Quality command will read the value for the Link_Quality for the specified Connection Handle.
Read_RSSI	The Read_RSSI command will read the value for the Received Signal Strength Indication (RSSI) for a connection handle to another Bluetooth device.

Bluetooth

4.9.1 Read_Failed_Contact_Counter

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Failed_ Contact_Counter	0x0001	Connection_Handle	Status, Connection_Handle, Failed_Contact_Counter

Description:

This command will read the value for the Failed_Contact_Counter parameter for a particular connection to another device. The Connection_Handle must be a Connection_Handle for an ACL connection. The Failed_Contact_Counter records the number of consecutive incidents in which either the slave or master didn't respond after the flush timeout had expired, and the L2CAP packet that was currently being transmitted was automatically 'flushed'. When this occurs, the Failed_Contact_Counter is incremented by 1. The Failed_Contact_ Counter for a connection is reset to zero on the following conditions:

1. When a new connection is established

2. When the Failed_Contact_Counter is > zero and an L2CAP packet is acknowledged for that connection

3. When the Reset_Failed_Contact_Counter command has been issued

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	The Connection Handle for the Connection for which the Failed Contact Counter should be read.
	Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Failed_Contact_Counter command succeeded.
0x01-0xFF	Read_Failed_Contact_Counter command failed. See Table 6.1 on page 745 for list of Error Codes.

Bluetooth

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	The Connection Handle for the Connection for which the Failed Contact Counter has been read.
	Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Failed_Contact_Counter: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	Number of consecutive failed contacts for a connection corresponding to the connection handle.

Event(s) generated (unless masked away):

When the Read_Failed_Contact_Counter command has completed, a Command Complete event will be generated.

Bluetooth

4.9.2 Reset_Failed_Contact_Counter

Command	OCF	Command Parameters	Return Parameters
HCI_Reset_Failed_ Contact_Counter	0x0002	Connection_Handle	Status, Connection_Handle

Description:

This command will reset the value for the Failed_Contact_Counter parameter
for a particular connection to another device. The Connection_Handle must be
a Connection_Handle for an ACL connection. The Failed_Contact_Counter
records the number of consecutive incidents in which either the slave or master
didn't respond after the flush timeout had expired, and the L2CAP packet that
was currently being transmitted was automatically 'flushed'. When this occurs,
the Failed_Contact_Counter is incremented by 1.The Failed_Contact_
Counter for a connection is reset to zero on the following conditions:

1. When a new connection is established

2. When the Failed_Contact_Counter is > zero and an L2CAP packet is
   acknowledged for that connection

3. When the Reset_Failed_Contact_Counter command has been issued

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	The Connection Handle for the Connection for which the Failed Contact Counter should be reset. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Reset_Failed_Contact_Counter command succeeded.
0x01-0xFF	Reset_Failed_Contact_Counter command failed. See Table 6.1 on page 745 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	The Connection Handle for the Connection for which the Failed Contact Counter has been reset.
	Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Event(s) generated (unless masked away):

When the Reset_Failed_Contact_Counter command has completed, a Command
Complete event will be generated.

Bluetooth

4.9.3 Get_Link_Quality

Command	OCF	Command Parameters	Return Parameters
HCI_Get_Link_Quality	0x0003	Connection_Handle	Status, Connection_Handle, Link_Quality

Description:

This command will return the value for the Link_Quality for the specified Connection Handle. The Connection_Handle must be a Connection_Handle for an ACL connection. This command will return a Link_Quality value from 0-255, which represents the quality of the link between two Bluetooth devices. The higher the value, the better the link quality is. Each Bluetooth module vendor will determine how to measure the link quality.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	The Connection Handle for the connection for which link quality parameters are to be read. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Get_Link_Quality command succeeded.
0x01-0xFF	Get_Link_Quality command failed.See Table 6.1 on page 745 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	The Connection Handle for the connection for which the link quality parameter has been read. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Bluetooth

Link_Quality: *Size: 1 Byte*

Value	Parameter Description
0xXX	The current quality of the Link connection between the local device and the remote device specified by the Connection Handle
	Range: 0x00 – 0xFF
	The higher the value, the better the link quality is.

Event(s) generated (unless masked away):

When the Get_Link_Quality command has completed, a Command Complete event will be generated.

4.9.4 Read_RSSI

Command	OCF	Command Parameters	Return Parameters
HCI_Read_RSSI	0x0005	Connection_Handle	Status, Connection_Handle,RSSI

Description:

This command will read the value for the difference between the measured Received Signal Strength Indication (RSSI) and the limits of the Golden Receive Power Range (see Radio Specification Section 4.7 on page 26) for a connection handle to another Bluetooth device. The Connection_Handle must be a Connection_Handle for an ACL connection. Any positive RSSI value returned by the Host 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 Bluetooth device is able to tell whether the RSSI is inside, above or below the Golden Device Power Range.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	The Connection Handle for the Connection for which the RSSI is to be read.
	Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_RSSI command succeeded.
0x01-0xFF	Read_RSSI command failed. See Table 6.1 on page 745 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	The Connection Handle for the Connection for which the RSSI has been read.
	Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Bluetooth

RSSI: *Size: 1 Byte*

Value	Parameter Description
N = 0xXX	Size: 1 Byte (signed integer) Range: -128 $\leq$ N $\leq$ 127 Units: dB

Event(s) generated (unless masked away):

When the Read_RSSI command has completed, a Command Complete event will be generated.

Bluetooth

4.10   TESTING COMMANDS

The Testing commands are used to provide the ability to test various functionalities 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

Command	Command Summary Description
Read_Loopback_Mode	The Read_Loopback_Mode will read the value for the setting of the Host Controllers Loopback Mode. The setting of the Loopback Mode will determine the path of information.
Write_Loopback_Mode	The Write_Loopback_Mode will write the value for the setting of the Host Controllers Loopback Mode. The setting of the Loopback Mode will determine the path of information.
Enable_Device_Under_Test_Mode	The Enable_Device_Under_Test_Mode command will allow the local Bluetooth module to enter test mode via LMP test commands. The Host issues this command when it wants the local device to be the DUT for the Testing scenarios as described in the Bluetooth Test Mode document.

Bluetooth

4.10.1 Read_Loopback_Mode

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Loopback_Mode	0x0001		Status, Loopback_Mode

Description:

This command will read the value for the setting of the Host Controller's Loopback Mode. The setting of the Loopback Mode will determine the path of information. In Non-testing Mode operation, the Loopback Mode 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 and SCO) and Command Packet that is sent from the Host to the Host Controller is sent back with no modifications by the Host Controller, as shown in Fig. 4.5 on page 697.

When the Bluetooth Host Controller enters Local Loopback Mode, it shall respond with four Connection Complete events, one for an ACL channel and three for SCO channels, so that the Host gets connection handles to use when sending ACL and SCO data. When in Local Loopback Mode the Host Controller loops back commands and data to the Host. The Loopback Command event is used to loop back commands that the Host sends to the Host Controller.

There are some commands that are not looped back in Local Loopback Mode: Reset, Set_Host_Controller_To_Host_Flow_Control, Host_Buffer_Size, Host_Number_Of_Completed_Packets, Read_Buffer_Size, Read_Loopback_Mode and Write_Loopback_Mode.These commands should be executed in the way they are normally executed. The commands Reset and Write_Loopback_Mode can be used to exit local loopback mode. If Write_Loopback_Mode is used to exit Local Loopback Mode, four Disconnection Complete events should be sent to the Host, corresponding to the Connection Complete events that were sent when entering Local Loopback Mode. 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 Host Controller will refuse incoming connection attempts. This allows the Host Controller Transport Layer to be tested without any other variables.

If a device is set to Remote Loopback Mode, it will send back all data (ACL and SCO) that comes over the air, and it will only allow a maximum of one ACL connection and three SCO connections – and these should be all to the same remote device. If there already are connections to more than one remote device and there is an attempt to set the local device to Remote Loopback Mode, the attempt will be refused. See Fig. 4.6 on page 697 where the rightmost device is set to Remote Loopback Mode and the leftmost device is set to

Bluetooth

Non-testing Mode. This allows the Bluetooth Air link to be tested without any other variables.

Figure 4.5: Local Loopback Mode

Figure 4.6: Remote Loopback Mode

Command Parameters:

None.

Bluetooth

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Loopback_Mode command succeeded.
0x01-0xFF	Read_Loopback_Mode command failed. See Table 2 on page 260 for list of Error Codes.

Loopback_Mode: *Size: 1 Byte*

Value	Parameter Description
0x00	No Loopback mode enabled. **Default.**
0x01	Enable Local Loopback.
0x02	Enable Remote Loopback.
0x03-0xFF	Reserved for Future Use.

Event(s) generated (unless masked away):

When the Read_Loopback_Mode command has completed, a Command Complete event will be generated.

Bluetooth

4.10.2 Write_Loopback_Mode

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Loopback_Mode	0x0002	Loopback_Mode	Status

Description:

This command will write the value for the setting of the Host Controller's Loopback Mode. The setting of the Loopback Mode will determine the path of information. In Non-testing Mode operation, the Loopback Mode 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 and SCO) and Command Packet that is sent from the Host to the Host Controller is sent back with no modifications by the Host Controller, as shown in Fig. 4.7 on page 700.

When the Bluetooth Host Controller enters Local Loopback Mode, it shall respond with four Connection Complete events, one for an ACL channel and three for SCO channels, so that the Host gets connection handles to use when sending ACL and SCO data. When in Local Loopback Mode, the Host Controller loops back commands and data to the Host. The Loopback Command event is used to loop back commands that the Host sends to the Host Controller.

There are some commands that are not looped back in Local Loopback Mode: Reset, Set_Host_Controller_To_Host_Flow_Control, Host_Buffer_Size, Host_Number_Of_Completed_Packets, Read_Buffer_Size, Read_Loopback_Mode and Write_Loopback_Mode. These commands should be executed in the way they are normally executed. The commands Reset and Write_Loopback_Mode can be used to exit local loopback mode.

If Write_Loopback_Mode is used to exit Local Loopback Mode, four Disconnection Complete events should be sent to the Host corresponding to the Connection Complete events that were sent when entering Local Loopback Mode. 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 Host Controller will refuse incoming connection attempts. This allows the Host Controller Transport Layer to be tested without any other variables.

If a device is set to Remote Loopback Mode, it will send back all data (ACL and SCO) that comes over the air. It will only allow a maximum of one ACL connection and three SCO connections, and these should all be to the same remote device. If there already are connections to more than one remote device and there is an attempt to set the local device to Remote Loopback Mode, the attempt will be refused.

See Fig. 4.8 on page 700, where the rightmost device is set to Remote Loop-back Mode and the leftmost device is set to Non-testing Mode. This allows the Bluetooth Air link to be tested without any other variables.
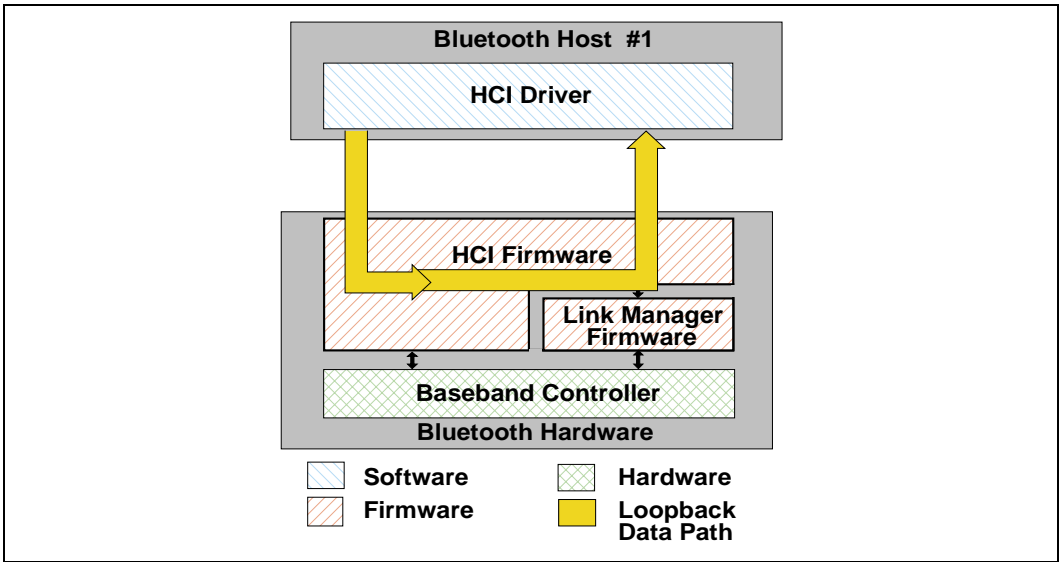
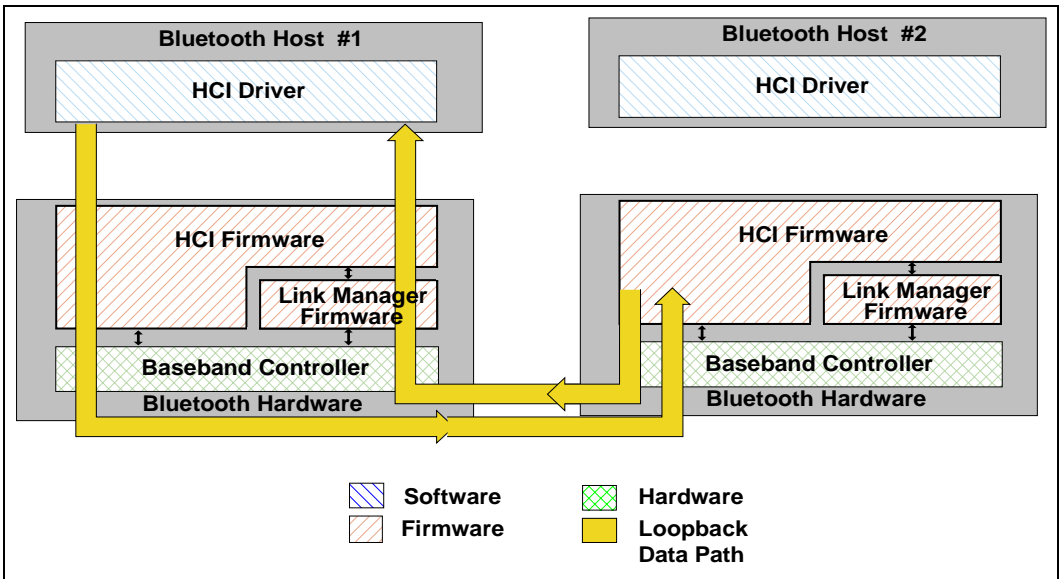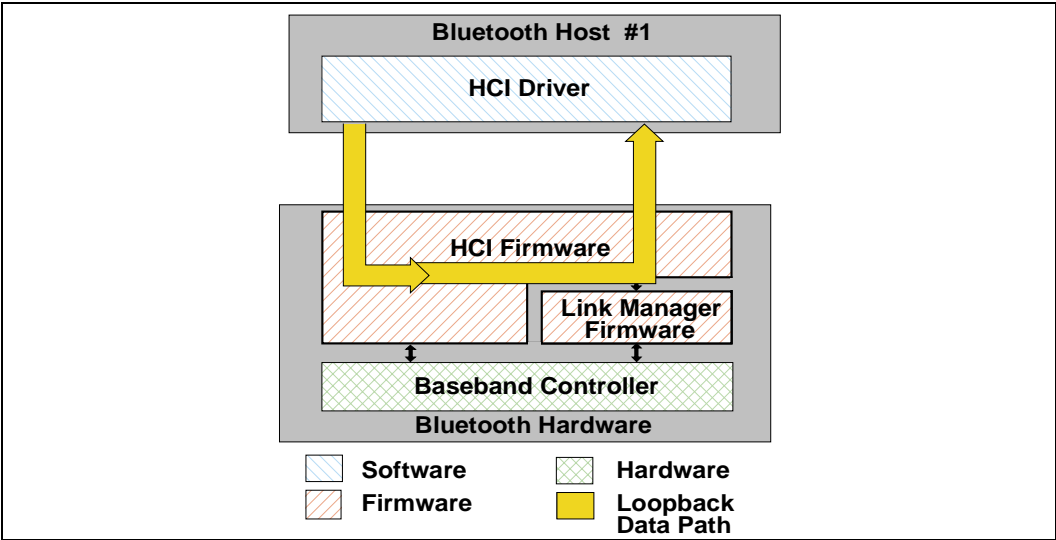

Figure 4.7: Local Loopback Mode
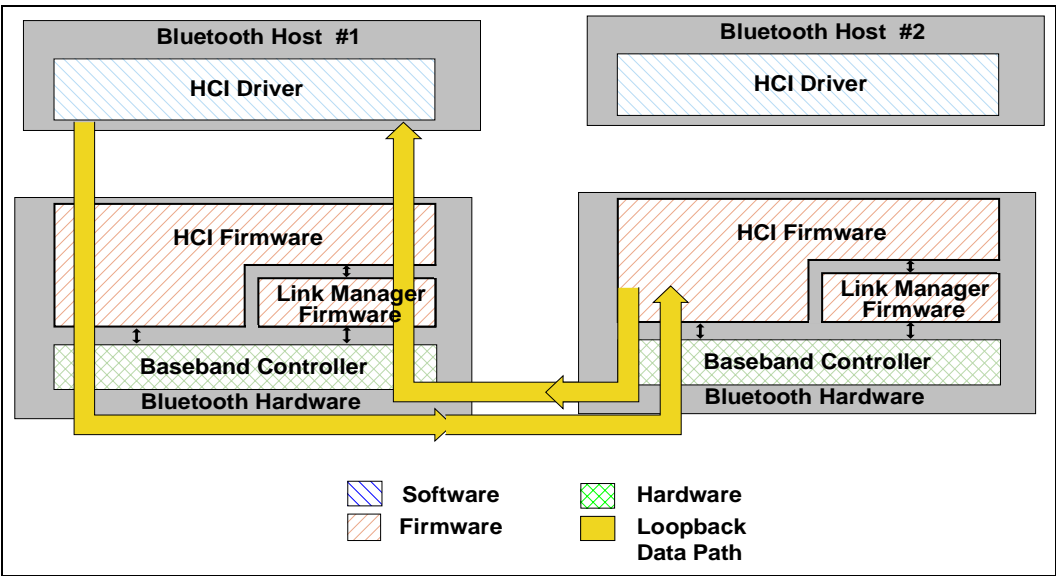
Figure 4.8: Remote Loopback Mode

Bluetooth

Command Parameters:

Loopback_Mode: *Size: 1 Byte*

Value	Parameter Description
0x00	No Loopback mode enabled. **Default.**
0x01	Enable Local Loopback.
0x02	Enable Remote Loopback.
0x03-0xFF	Reserved for Future Use.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write_Loopback_Mode command succeeded.
0x01-0xFF	Write_Loopback_Mode command failed. See Table 6.1 on page 745 for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Loopback_Mode command has completed, a Command Complete event will be generated.

4.10.3 Enable_Device_Under_Test_Mode

Command	OCF	Command Parameters	Return Parameters
HCI_Enable_Device_Under_Test_Mode	0x0003		Status

Description:

The Enable_Device_Under_Test_Mode command will allow the local Bluetooth module to enter test mode via LMP test commands. For details see "Link Manager Protocol" on page 185. The Host issues this command when it wants the local device to be the DUT for the Testing scenarios as described in the "Bluetooth Test Mode" on page 803. When the Host Controller receives this command, it will complete the command with a Command Complete event. The Host 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 can issue the HCI_Reset command. This command prevents remote Bluetooth devices from causing the local Bluetooth device to enter test mode without first issuing this command.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Enter_Device_Under_Test_Mode command succeeded.
0x01-0xFF	Enter_Device_Under_Test_Mode command failed. See Table 6.1 on page 745 for list of Error Codes.

Event(s) generated (unless masked away):

When the Enter_Device_Under_Test_Mode command has completed, a Command Complete event will be generated.

5 EVENTS

5.1 EVENT

In addition to the events listed below, event code 0xFF is reserved for the event code used for vendor-specific debug events, and event code 0xFE is reserved for Bluetooth Logo Testing.

Event	Event Summary Description
Inquiry Complete event	The Inquiry Complete event indicates that the Inquiry is finished.
Inquiry Result event	The Inquiry Result event indicates that a Bluetooth device or multiple Bluetooth devices have responded so far during the current Inquiry process.
Connection Complete event	The Connection Complete event indicates to both of the Hosts forming the connection that a new connection has been established.
Connection Request event	The Connection Request event is used to indicate that a new incoming connection is trying to be established.
Disconnection Complete event	The Disconnection Complete event occurs when a connection has been terminated.
Authentication Complete event	The Authentication Complete event occurs when authentication has been completed for the specified connection.
Remote Name Request Complete event	The Remote Name Request Complete event is used to indicate a remote name request has been completed. The Remote_Name event parameter is a UTF-8 encoded string with up to 248 bytes in length.
Encryption Change event	The Encryption Change event is used to indicate that the change in the encryption has been completed for the Connection Handle specified by the Connection_Handle event parameter.
Change Connection Link Key Complete event	The Change Connection Link Key Complete event is used to indicate that the change in the Link Key for the Connection Handle specified by the Connection_Handle event parameter had been completed.
Master Link Key Complete event	The Master Link Key Complete event is used to indicate that the change in the temporary Link Key or in the semi-permanent link keys on the Bluetooth master side has been completed.
Read Remote Supported Features Complete event	The 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 Bluetooth device specified by the Connection_Handle event parameter.

Table 5.1: List of Supported Events

Bluetooth.

Event	Event Summary Description
Read Remote Version Information Complete event	The Read Remote Version Information Complete event is used to indicate the completion of the process of the Link Manager obtaining the version information of the remote Bluetooth device specified by the Connection_Handle event parameter.
QoS Setup Complete event	The QoS Setup Complete event is used to indicate the completion of the process of the Link Manager setting up QoS with the remote Bluetooth device specified by the Connection_Handle event parameter.
Command Complete event	The Command Complete event is used by the Host Controller to pass the return status of a command and the other event parameters for each HCI Command.
Command Status event	The Command Status event is used to indicate that the command described by the Command_Opcode parameter has been received and the Host Controller is currently performing the task for this command.
Hardware Error event	The Error event is used to indicate some type of hardware failure for the Bluetooth device.
Flush Occurred event	The Flush Occurred event is used to indicate that, for the specified Connection Handle, the current user data to be transmitted has been removed.
Role Change event	The Role Change event is used to indicate that the current Bluetooth role related to the particular connection has been changed.
Number Of Completed Packets event	The Number Of Completed Packets event is used by the Host Controller to indicate to the Host how many HCI Data Packets have been completed for each Connection Handle since the previous Number Of Completed Packets event was sent.
Mode Change event	The Mode Change event is used to indicate when the device associated with the Connection Handle changes between Active, Hold, Sniff and Park mode.
Return Link Keys event	The Return Link Keys event is used to return stored link keys after a Read_Stored_Link_Key command is used.
PIN Code Request event	The PIN Code Request event is used to indicate that a PIN code is required to create a new link key for a connection.
Link Key Request event	The Link Key Request event is used to indicate that a Link Key is required for the connection with the device specified in BD_ADDR.
Link Key Notification event	The 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.
Loopback Command event	The Loopback Command event is used to loop back most commands that the Host sends to the Host Controller.

Table 5.1: List of Supported Events

Bluetooth

Event	Event Summary Description
Data Buffer Overflow event	The Data Buffer Overflow event is used to indicate that the Host Controller's data buffers have overflowed, because the Host has sent more packets than allowed.
Max Slots Change event	This event is used to notify the Host about the LMP_Max_Slots parameter when the value of this parameter changes.
Read Clock Offset Complete event	The Read Clock Offset Complete event is used to indicate the completion of the process of the LM obtaining the Clock offset information.
Connection Packet Type Changed event	The Connection Packet Type Changed event is used to indicate the completion of the process of the Link Manager changing the Packet Types used for the specified Connection_Handle.
QoS Violation event	The QoS Violation event is used to indicate the Link Manager is unable to provide the current QoS requirement for the Connection Handle.
Page Scan Mode Change event	The Page Scan Mode Change event indicates that the connected remote Bluetooth device with the specified Connection_Handle has successfully changed the Page_Scan_Mode.
Page Scan Repetition Mode Change event	The Page Scan Repetition Mode Change event indicates that the connected remote Bluetooth device with the specified Connection_Handle has successfully changed the Page_Scan_Repetition_Mode (SR).

Table 5.1: List of Supported Events

Bluetooth

5.2 POSSIBLE EVENTS

The events provide a method to return parameters and data associated for each event.

5.2.1 Inquiry Complete event

Event	Event Code	Event Parameters
Inquiry Complete	0x01	Status, Num_Responses

Description:

The 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. In addition, the Num_Responses parameter contains the number of Bluetooth devices, which responded during the latest inquiry.

Event Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Inquiry command completed successfully.
0x01-0xFF	Inquiry command failed. See Table 6.1 on page 745 for list of Error Codes.

Num_Responses: *Size: 1 Byte*

Value	Parameter Description
0xXX	Number of responses from the Inquiry.

Bluetooth

5.2.2 Inquiry Result event

Event	Event Code	Event Parameters
Inquiry Result	0x02	Num_Responses, BD_ADDR[i], Page_Scan_Repetition_Mode[i], Page_Scan_Period_Mode[i], Page_Scan_Mode[i], Class_of_Device[i] Clock_Offset[i]

Description:

The Inquiry Result event indicates that a Bluetooth device or multiple Bluetooth devices have responded so far during the current Inquiry process. This event will be sent from the Host 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 Host Controller may queue these Inquiry Responses and send multiple Bluetooth devices information in one Inquiry Result event. The event can be used to return one or more Inquiry responses in one event. This event contains the BD_ADDR, Page_Scan_Repetition_Mode, Page_Scan_Period_Mode, Page_Scan_Mode, Clock_Offset, and the Class of Device for each Bluetooth device that responded to the latest inquiry.

Event Parameters:

Num_Responses: *Size: 1 Byte*

Value	Parameter Description
0xXX	Number of responses from the Inquiry.

BD_ADDR[i]:   *Size: 6 Bytes * Num_Responses*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR for each device which responded.

Bluetooth

Page_Scan_Repetition_Mode[i]:   *Size: 1 Byte* Num_Responses*

Value	Parameter Description
0x00	R0
0x01	R1
0x02	R2
0x03 – 0xFF	Reserved

Page_Scan_Period_Mode[i]:   *Size: 1 Byte* Num_Responses*

Value	Parameter Description
0x00	P0
0x01	P1
0x02	P2
0x03 – 0xFF	Reserved

Page_Scan_Mode[i]:   *Size: 1 Byte* Num_Responses*

Value	Parameter Description
0x00	Mandatory Page Scan Mode
0x01	Optional Page Scan Mode I
0x02	Optional Page Scan Mode II
0x03	Optional Page Scan Mode III
0x04 – 0xFF	Reserved

Class_of_Device[i]:   *Size: 3 Bytes * Num_Responses*

Value	Parameter Description
0xXXXXXX	Class of Device for the device

Clock_Offset[i]:   *Size: 2 Bytes * Num_Responses*

Bit format	Parameter Description
Bit 14.0	Bit 16.2 of CLKslave-CLKmaster.
Bit 15	Reserved

Bluetooth

5.2.3 Connection Complete event

Event	Event Code	Event Parameters
Connection Complete	0x03	Status, Connection_Handle, BD_ADDR, Link_Type, Encryption_Mode

Description:

The 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 Create Connection, Add_SCO_Connection, or Accept_Connection_Request or Reject_Connection_Request command and then received a Command Status event, if the issued command failed or was successful.

Event Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Connection successfully completed.
0x01-0xFF	Connection failed to Complete. See Table 6.1 on page 745 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify a connection between to Bluetooth devices. The Connection Handle is used as an identifier for transmitting and receiving voice or data.
	Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the other connected Device forming the connection.

Bluetooth

Link_Type: *Size: 1 Byte*

Value	Parameter Description
0x00	SCO connection (Voice Channels).
0x01	ACL connection (Data Channels).
0x02-0xFF	Reserved for Future Use.

Encryption_Mode: *Size: 1 Byte*

Value	Parameter Description
0x00	Encryption disabled.
0x01	Encryption only for point-to-point packets.
0x02	Encryption for both point-to-point and broadcast packets.
0x03-0xFF	Reserved.

Bluetooth

5.2.4 Connection Request event

Event	Event Code	Event Parameters
Connection Request	0x04	BD_ADDR, Class_of_Device, Link_Type

Description:

The 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. If this event is masked away and there is an incoming connection attempt and the Host Controller is not set to auto-accept this connection attempt, the Host Controller will automatically refuse the connection attempt. When the Host receives this event, it should respond with either an Accept_Connection_Request or Reject_Connection_Request command before the timer Conn_Accept_Timeout expires.

Event Parameters:

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the device that requests the connection.

Class_of_Device: *Size: 3 Bytes*

Value	Parameter Description
0xXXXXXX	Class of Device for the device, which request the connection.

Link_Type: *Size: 1 Byte*

Value	Parameter Description
0x00	SCO connection requested (Voice Channels).
0x01	ACL connection requested (Data Channels).
0x02-0xFF	Reserved for Future Use.

Bluetooth

5.2.5 Disconnection Complete event

Event	Event Code	Event Parameters
Disconnection Complete	0x05	Status, Connection_Handle, Reason

Description:

The 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 can be ignored by the Host. For example, this can be the case if the Host has issued the 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.
Note: When a physical link fails, one 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 Byte*

Value	Parameter Description
0x00	Disconnection has occurred.
0x01-0xFF	Disconnection failed to complete. See Table 6.1 on page 745 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle which was disconnected. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Reason: *Size: 1 Byte*

Value	Parameter Description
0x08, 0x13-0x16, 0x1A	Connection Timeout (0x08), Other End Terminated Connection error codes (0x13-0x15), Connection Terminated by Local Host (0x16), and Unsupported Remote Feature error code (0x1A). See Table 6.1 on page 745 for list of Error Codes.

Bluetooth

5.2.6 Authentication Complete event

Event	Event Code	Event Parameters
Authentication Complete	0x06	Status, Connection_Handle

Description:

The Authentication Complete event occurs when authentication has been completed for the specified connection. The Connection_Handle must be a Connection_Handle for an ACL connection.

Event Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Authentication Request successfully completed.
0x01-0xFF	Authentication Request failed to complete. See Table 6.1 on page 745 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle for which Authentication has been performed.
	Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Bluetooth

5.2.7 Remote Name Request Complete event

Event	Event Code	Event Parameters
Remote Name Request Complete	0x07	Status, BD_ADDR, Remote_Name

Description:

The 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 up to 248 bytes in length. The Remote_Name event parameter will be null-terminated (0x00) if the UTF-8 encoded string is less than 248 bytes. The BD_ADDR event parameter is used to identify which device the user-friendly name was obtained from.
Note: the Remote_Name Parameter is received starting with the first byte of the name. This is an exception to the Little Endian order format for transmitting multi-byte parameters.

Event Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Remote_Name_Request command succeeded.
0x01-0xFF	Remote_Name_Request command failed. See Table 6.1 on page 745 for list of Error Codes.

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR for the device whose name was requested.

Remote_Name: *Size: 248 Bytes*

Value	Parameter Description
Name[248]	A UTF-8 encoded user-friendly descriptive name for the remote device. A UTF-8 encoded name can be up to 248 bytes in length. If it is shorter than 248 bytes, the end is indicated by a NULL byte (0x00).

5.2.8 Encryption Change event

Event	Event Code	Event Parameters
Encryption Change	0x08	Status, Connection_Handle, Encryption_Enable

Description:

The Encryption Change event is used to indicate that the change in the encryption has been completed for the Connection Handle specified by the Connection_Handle event parameter. The Connection_Handle will be a Connection_Handle for an ACL connection. The Encryption_Enable event parameter specifies the new Encryption Enable for the Connection Handle specified by the Connection_Handle event parameter. This event will occur on both devices to notify both Hosts when Encryption has changed for the specified Connection Handle between two devices.

Event Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Encryption Change has occurred.
0x01-0xFF	Encryption Change failed. See Table 6.1 on page 745 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle for which the link layer encryption has been enabled/disabled for all Connection Handles with the same Bluetooth device endpoint as the specified Connection Handle. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Encryption_Enable: *Size: 1 Byte*

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

5.2.9 Change Connection Link Key Complete event

Event	Event Code	Event Parameters
Change Connection Link Key Complete	0x09	Status, Connection_Handle

Description:

The Change Connection Link Key Complete event is used to indicate that the change in the Link Key for the Connection Handle specified by the Connection_Handle event parameter has been completed.
The Connection_Handle will be a Connection_Handle for an ACL connection.
The Change Connection Link Key Complete event is sent only to the Host which issued the Change_Connection_Link_Key command.

Event Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Change_Connection_Link_Key command succeeded.
0x01-0xFF	Change_Connection_Link_Key command failed. See Table 6.1 on page 745 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle which the Link Key has been change for all Connection Handles with the same Bluetooth device end point as the specified Connection Handle.
	Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

5.2.10 Master Link Key Complete event

Event	Event Code	Event Parameters
Master Link Key Complete	0x0A	Status, Connection_Handle, Key_Flag

Description:

The Master Link Key Complete event is used to indicate that the Link Key managed by the master of the piconet has been changed. The Connection_Handle will be a Connection_Handle for an ACL connection. The link key used for the connection will be the temporary link key of the master device 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 of the Master, or the semi-permanent link keys) is now being used in the piconet.
Note: for a master, the change from a semi-permanent Link Key to temporary Link Key will affect all Connection Handles related to the piconet. For a slave, this change affects only this particular connection handle. A temporary link key must be used when both broadcast and point-to-point traffic shall be encrypted.

Event Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Master_Link_Key command succeeded.
0x01-0xFF	Master_Link_Key command failed. See Table 6.1 on page 745 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle for which the Link Key has been changed for all devices in the same piconet.
	Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Key_Flag: *Size: 1 Byte*

Value	Parameter Description
0x00	Using Semi-permanent Link Key.
0x01	Using Temporary Link Key.

Bluetooth

5.2.11 Read Remote Supported Features Complete event

Event	Event Code	Event Parameters
Read Remote Supported Features Complete	0x0B	Status, Connection_Handle, LMP_Features

Description:

The 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 Bluetooth 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 list of LMP features. For details see "Link Manager Protocol" on page 185.

Event Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Remote_Supported_Features command succeeded.
0x01-0xFF	Read_Remote_Supported_Features command failed. See Table 6.1 on page 745 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle which is used for the Read_Remote_Supported_Features command. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

LMP_Features: *Size: 8 Bytes*

Value	Parameter Description
0xXXXXXXXX XXXXXXXX	Bit Mask List of LMP features. See "Link Manager Protocol" on page 185.

Bluetooth

5.2.12 Read Remote Version Information Complete event

Event	Event Code	Event Parameters
Read Remote Version Information Complete	0x0C	Status, Connection_Handle, LMP_Version, Manufacturer_Name, LMP_Subversion

Description:

The Read Remote Version Information Complete event is used to indicate the completion of the process of the Link Manager obtaining the version information of the remote Bluetooth device specified by the Connection_Handle event parameter. The Connection_Handle will be a Connection_Handle for an ACL connection. The LMP_Version event parameter defines the major hardware version of the Bluetooth hardware. This event parameter only changes when new versions of the Bluetooth hardware are produced for new Bluetooth SIG specifications; it is controlled by the SIG. The Manufacturer_Name event parameter indicates the manufacturer of the remote Bluetooth module. The LMP_
Subversion event parameter should be controlled by the manufacturer and should be changed as needed. The LMP_Subversion event parameter defines the various revisions that each version of the Bluetooth hardware will go through as design processes change and errors are fixed. This allows the software to determine what Bluetooth hardware is being used and, if necessary, to work around various bugs in the hardware.

Event Parameters:

Status: Size: 1 Byte

Value	Parameter Description
0x00	Read_Remote_Version_Information command succeeded.
0x01-0xFF	Read_Remote_Version_Information command failed. See Table 6.1 on page 745 for list of Error Codes.

Connection_Handle: Size: 2 Bytes (12 Bits meaningful)

Value	Parameter Description
0xXXXX	Connection Handle which is used for the Read_Remote_Version_Information command. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Bluetooth

LMP_Version: *Size: 1 Byte*

Value	Parameter Description
0xXX	Version of the Current LMP in the remote Bluetooth Hardware, see Table 5.2 on page 231 in the Link Manager Protocol for assigned values (VersNr).

Manufacturer_Name: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	Manufacturer Name of the remote Bluetooth Hardware, see Table 5.2 on page 231 in the Link Manager Protocol for assigned values (CompId).

LMP_Subversion: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	Subversion of the Current LMP in the remote Bluetooth Hardware, see Table 5.2 on page 231 in the Link Manager Protocol for assigned values (SubVersNr).

Bluetooth

5.2.13 QoS Setup Complete event

Event	Event Code	Event Parameters
QoS Setup Complete	0x0D	Status, Connection_Handle, Flags, Service_Type, Token_Rate, Peak_Bandwidth, Latency, Delay_Variation

Description:

The QoS Setup Complete event is used to indicate the completion of the process of the Link Manager setting up QoS with the remote Bluetooth device specified by the Connection_Handle event parameter. The Connection_Handle will be a Connection_Handle for an ACL connection. For more detail see "Logical Link Control and Adaptation Protocol Specification" on page 245.

Event Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	QoS_Setup command succeeded.
0x01-0xFF	QoS_Setup command failed. See Table 6.1 on page 745 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle which is used for the QoS_Setup command. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Flags: *Size: 1 Byte*

Value	Parameter Description
0x00 – 0xFF	Reserved for Future Use.

Bluetooth

Service_Type: *Size: 1 Byte*

Value	Parameter Description
0x00	No Traffic Available.
0x01	Best Effort Available.
0x02	Guaranteed Available.
0x03-0xFF	Reserved for Future Use.

Token_Rate: *Size: 4 Bytes*

Value	Parameter Description
0xXXXXXXXX	Available Token Rate, in bytes per second.

Peak_Bandwidth: *Size: 4 Bytes*

Value	Parameter Description
0xXXXXXXXX	Available Peak Bandwidth, in bytes per second.

Latency: *Size: 4 Bytes*

Value	Parameter Description
0xXXXXXXXX	Available Latency, in microseconds.

Delay_Variation: *Size: 4 Bytes*

Value	Parameter Description
0xXXXXXXXX	Available Delay Variation, in microseconds.

Bluetooth

5.2.14 Command Complete event

Event	Event Code	Event Parameters
Command Complete	0x0E	Num_HCI_Command_Packets, Command_Opcode, Return_Parameters

Description:

The Command Complete event is used by the Host 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 Host Controller to indicate the number of HCI command packets the Host can send to the Host Controller. If the Host 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 Host Controller is ready to receive HCI command packets, the Host Controller generates a Command Complete event with the Command_Opcode 0x0000, and the Num_HCI_Command_Packets event parameter is set to 1 or more. Command_Opcode, 0x0000 is a NOP (No OPeration), and can be used 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 Byte*

Value	Parameter Description
N = 0xXX	The Number of HCI command packets which are allowed to be sent to the Host Controller from the Host. Range for N: 0 – 255

Command_Opcode: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	Opcode of the command which caused this event.

Return_Parameter(s):   *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.

Bluetooth

5.2.15 Command Status event

Event	Event Code	Event Parameters
Command Status	0x0F	Status, Num_HCI_Command_Packets, Command_Opcode

Description:

The Command Status event is used to indicate that the command described by the Command_Opcode parameter has been received, and that the Host Controller is currently performing the task for this command. This event is needed to provide mechanisms for asynchronous operation, which makes it possible to prevent the Host from waiting for a command to finish. If the command can not 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 Host Controller to indicate the number of HCI command packets the Host can send to the Host Controller. If the Host 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 Host Controller is ready to receive HCI command packets, the Host Controller generates a Command Status event with Status 0x00 and Command_Opcode 0x0000, and the Num_HCI_Command_Packets event parameter is set to 1 or more. Command_Opcode, 0x0000 is a NOP (No OPeration) and can be used to change the number of outstanding HCI command packets that the Host can send before waiting.

Event Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Command currently in pending.
0x01-0xFF	Command failed. See Table 6.1 on page 745 for list of Error Codes.

Num_HCI_Command_Packets: *Size: 1 Byte*

Value	Parameter Description
N = 0xXX	The Number of HCI command packets which are allowed to be sent to the Host Controller from the Host. Range for N: 0 – 255

Command_Opcode: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	Opcode of the command which caused this event and is pending completion.

Bluetooth

5.2.16 Hardware Error event

Event	Event Code	Event Parameters
Hardware Error	0x10	Hardware_Code

Description:

The Hardware Error event is used to indicate some type of hardware failure for the Bluetooth device. This event is used to notify the Host that a hardware failure has occurred in the Bluetooth module.

Event Parameters:

Hardware_Code: *Size: 1 Byte*

Value	Parameter Description
0x00	These Hardware_Codes will be implementation-specific, and will be assigned to indicate various hardware problems.

5.2.17 Flush Occurred event

Event	Event Code	Event Parameters
Flush Occurred	0x11	Connection_Handle

Description:

The Flush Occurred event is used to indicate that, for the specified Connection Handle, the current user data to be transmitted has been removed. The Connection_Handle will be a Connection_Handle for an ACL connection. This could result from the flush command, or be due to the automatic flush. Multiple blocks of an L2CAP packet could have been pending in the Host Controller. If one baseband packet part of an L2CAP packet is flushed, then the rest of the HCI data packets for the L2CAP packet must also be flushed.

Event Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle which was flushed.
	Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Bluetooth.

5.2.18 Role Change event

Event	Event Code	Event Parameters
Role Change	0x12	Status, BD_ADDR, New_Role

Description:

The Role Change event is used to indicate that the current Bluetooth role related to the particular connection has changed. This event only occurs when both the remote and local Bluetooth devices have completed their role change for the Bluetooth device associated with the BD_ADDR event parameter. This event allows both affected Hosts to be notified when the Role has been changed.

Event Parameters:

Status: Size: 1 Byte

Value	Parameter Description
0x00	Role change has occurred.
0x01-0xFF	Role change failed. See Table 6.1 on page 745 for list of Error Codes.

BD_ADDR: Size: 6 Bytes

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device for which a role change has completed.

New_Role: Size: 1 Byte

Value	Parameter Description
0x00	Currently the Master for specified BD_ADDR.
0x01	Currently the Slave for specified BD_ADDR.

Bluetooth

5.2.19 Number Of Completed Packets event

Event	Event Code	Event Parameters
Number Of Completed Packets	0x13	Number_of_Handles, Connection_Handle[i], HC_Num_Of_Completed_Packets[i]

Description:

The Number Of Completed Packets event is used by the Host Controller to indicate to the Host how many HCI Data Packets have been completed (transmitted or flushed) for each Connection Handle since the previous Number Of Completed Packets event was sent to the Host. This means that the corresponding buffer space has been freed in the Host Controller. Based on this information, and the HC_Total_Num_ACL_Data_Packets and HC_Total_ Num_SCO_Data_Packets return parameter of the Read_Buffer_Size command, the Host can determine for which Connection Handles the following HCI Data Packets should be sent to the Host Controller.The Number Of Completed Packets event must not be sent before the corresponding Connection Complete event. While the Host Controller has HCI data packets in its buffer, it must keep sending the Number Of Completed Packets event to the Host at least periodically, until it finally reports that all the pending ACL Data Packets have been transmitted or flushed. The rate with which this event is sent is manufacturer specific.
Note that Number Of Completed Packets events will not report on SCO connection handles if SCO Flow Control is disabled. (See Read/ Write_SCO_Flow_Control_Enable on and .)

Event Parameters:

Number_of_Handles: *Size: 1 Byte*

Value	Parameter Description
0xXX	The number of Connection Handles and Num_HCI_Data_Packets parameters pairs contained in this event. Range: 0-255

*Connection_Handle[i]:Size: Number_of_Handles * 2 Bytes(12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Bluetooth

*HC_Num_Of_Completed_Packets [i]:   Size: Number_of_Handles * 2 Bytes*

Value	Parameter Description
N = 0xXXXX	The number of HCI Data Packets that have been completed (transmitted or flushed) for the associated Connection Handle since the previous time the event was returned. Range for N: 0x0000-0xFFFF

Bluetooth.

5.2.20 Mode Change event

Event	Event Code	Event Parameters
Mode Change	0x14	Status, Connection_Handle, Current_Mode, Interval

Description:

The Mode Change event is used to indicate when the device associated with the Connection Handle changes between Active, Hold, Sniff and Park 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 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 Host Controller that is associated with the Connection Handle which has changed Modes will send the Mode Change event to its Host.

Event Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	A Mode Change has occurred.
0x01-0xFF	Hold_Mode, Sniff_Mode, Exit_Sniff_Mode, Park_Mode, or Exit_Park_Mode command failed. See Table 6.1 on page 745 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes(12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Bluetooth.

Current_Mode: *Size: 1 Byte*

Value	Parameter Description
0x00	Active Mode.
0x01	Hold Mode.
0x02	Sniff Mode.
0x03	Park Mode.
0x04-0xFF	Reserved for future use.

Interval: *Size: 2 Bytes*

Value	Parameter Description
N = 0xXXXX	Hold: Number of Baseband slots to wait in Hold Mode. Hold Interval = N * 0.625 msec (1 Baseband slot) Range for N: 0x0000-0xFFFF Time Range: 0-40.9 sec Sniff: Number of Baseband slots between sniff intervals. Time between sniff intervals = 0.625 msec (1 Baseband slot) Range for N: 0x0000-0xFFFF Time Range: 0-40.9 sec Park: Number of Baseband slots between consecutive beacons. Interval Length = N * 0.625 msec (1 Baseband slot) Range for N: 0x0000-0xFFFF Time Range: 0-40.9 Seconds

Bluetooth

5.2.21 Return Link Keys event

Event	Event Code	Event Parameters
Return Link Keys	0x15	Num_Keys, BD_ADDR [i], Link_Key[i]

Description:

The Return Link Keys event is used by the Host Controller to send the Host one or more stored Link Keys. Zero or more instances of this event will occur after the Read_Stored_Link_Key command. When there are no link keys stored, no Return Link Keys events will be returned. When there are link keys stored, the number of link keys returned in each Return Link Keys event is implementation specific.

Event Parameters:

Num_Keys: *Size: 1 Byte*

Value	Parameter Description
0xXX	Number of Link Keys contained in this event. Range: 0x01 – 0xFF

BD_ADDR [i]:   *Size: 6 Bytes * Num_Keys*

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

Link_Key[i]:   *Size:16 Bytes * Num_Keys*

Value	Parameter Description
0xXXXXXXXXXX XXXXXXXXXX XXXXXXXXXX	Link Key for the associated BD_ADDR.

Bluetooth

5.2.22 PIN Code Request event

Event	Event Code	Event Parameters
PIN Code Request	0x16	BD_ADDR

Description:

The PIN Code Request event is used to indicate that a PIN code is required to create a new link key. The Host must respond using either the PIN Code Request Reply or the PIN Code Request Negative Reply command, depending on whether the Host can provide the Host Controller with a PIN code or not. Note: If the PIN Code Request event is masked away, then the Host Controller will assume that the Host has no PIN Code.

When the Host Controller generates a 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 a Create_Connection or Authentication_Requested command from the remote Host), the local Host must respond with either a PIN_Code_Request_Reply or PIN_Code_Request_Negative_Reply command before the remote Link Manager detects LMP response timeout. (See "Link Manager Protocol" on page 185.)

Event Parameters:

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device which a new link key is being created for.

Bluetooth

5.2.23 Link Key Request event

Event	Event Code	Event Parameters
Link Key Request	0x17	BD_ADDR

Description:

The 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 will pass the requested Key to the Host Controller using the Link_Key_Request_Reply Command. If the Host does not have the requested stored Link Key, then the Host will use the Link_Key_Request_Negative_Reply Command to indicate to the Host Controller that the Host does not have the requested key.
Note: If the Link Key Request event is masked away, then the Host Controller will assume that the Host has no additional link keys.

When the Host Controller generates a 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 a Create_Connection or Authentication_Requested command from the remote Host), the local Host must respond with either a Link_Key_Request_Reply or Link_Key_Request_Negative_Reply command before the remote Link Manager detects LMP response timeout. (See "Link Manager Protocol" on page 185.)

Event Parameters:

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device which a stored link key is being requested.

Bluetooth

5.2.24 Link Key Notification event

Event	Event Code	Event Parameters
Link Key Notification	0x18	BD_ADDR, Link_Key

Description:

The 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 decided to store the Link Key in the Host Controller's Link Key Storage by using the Write_Stored_Link_Key command.

Event Parameters:

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device for which the new link key has been generated.

Link_Key: *Size: 16 Bytes*

Value	Parameter Description
0xXXXXXXXXXX XXXXXXXXXX XXXXXXXXXX	Link Key for the associated BD_ADDR.

Bluetooth

5.2.25 Loopback Command event

Event	Event Code	Event Parameters
Loopback Command	0x19	HCI_Command_Packet

Description:

When in Local Loopback mode, the Host Controller loops back commands and data to the Host. The Loopback Command event is used to loop back all commands that the Host sends to the Host Controller with some exceptions. See Section 4.10.1, "Read_Loopback_Mode," on page 696 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 bytes in the payload; since an HCI Command Packet has 3 bytes of header data, only the first 252 bytes 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.

Bluetooth

5.2.26 Data Buffer Overflow event

Event	Event Code	Event Parameters
Data Buffer Overflow	0x1A	Link_Type

Description:

This event is used to indicate that the Host 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 that the overflow was caused by ACL or SCO data.

Event Parameters:

Link_Type: *Size: 1 Byte*

Value	Parameter Description
0x00	SCO Buffer Overflow (Voice Channels).
0x01	ACL Buffer Overflow (Data Channels).
0x02-0xFF	Reserved for Future Use.

Bluetooth

5.2.27 Max Slots Change event

Event	Event Code	Event Parameters
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 will be sent each time the value of the LMP_Max_Slots parameter changes, as long as there is at least one connection to another device. The Connection_Handle will be a Connection_Handle for an ACL connection.

Event Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

LMP_Max_Slots: *Size: 1 byte*

Value	Parameter Description
0xXX	Maximum number of slots allowed to use for baseband packets, see "Link Manager Protocol" on page 185.

Bluetooth

5.2.28 Read Clock Offset Complete event

Event	Event Code	Event Parameters
Read Clock Offset Complete	0x1C	Status, Connection_Handle, Clock_Offset

Description:

The 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 Bluetooth device specified by the Connection_Handle event parameter. The Connection_Handle will be a Connection_Handle for an ACL connection.

Event Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Clock_Offset command succeeded.
0x01-0xFF	Read_Clock_Offset command failed. See Table 6.1 on page 745 for list of Error Codes.

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

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's Clock Offset parameter is returned. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Clock_Offset: *Size: 2 Bytes*

Bit format	Parameter Description
Bit 14.0	Bit 16.2 of CLKslave-CLKmaster.
Bit 15	Reserved.

Bluetooth

5.2.29 Connection Packet Type Changed event

Event	Event Code	Event Parameters
Connection Packet Type Changed	0x1D	Status, Connection_Handle, Packet_Type

Description:

The 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 Byte*

Value	Parameter Description
0x00	Connection Packet Type changed successfully.
0x01-0xFF	Connection Packet Type Changed failed. See Table 6.1 on page 745 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Packet_Type: *Size: 2 Bytes*
For ACL_Link_Type

Value	Parameter Description
0x0001	Reserved for future use.
0x0002	Reserved for future use.
0x0004	Reserved for future use.
0x0008	DM1
0x0010	DH1
0x0020	Reserved for future use.
0x0040	Reserved for future use.
0x0080	Reserved for future use.

Value	Parameter Description
0x0100	Reserved for future use.
0x0200	Reserved for future use.
0x0400	DM3
0x0800	DH3
0x1000	Reserved for future use.
0x2000	Reserved for future use.
0x4000	DM5
0x8000	DH5

For SCO_Link_Type

Value	Parameter Description
0x0001	Reserved for future use.
0x0002	Reserved for future use.
0x0004	Reserved for future use.
0x0008	Reserved for future use.
0x0010	Reserved for future use.
0x0020	HV1
0x0040	HV2
0x0080	HV3
0x0100	Reserved for future use.
0x0200	Reserved for future use.
0x0400	Reserved for future use.
0x0800	Reserved for future use.
0x1000	Reserved for future use.
0x2000	Reserved for future use.
0x4000	Reserved for future use.
0x8000	Reserved for future use.

5.2.30 QoS Violation event

Event	Event Code	Event Parameters
QoS Violation	0x1E	Connection_Handle

Description:

The QoS Violation event is used to indicate the Link Manager is unable to pro-
vide the current QoS requirement for the Connection Handle. This event indi-
cates that the Link Manager is unable to provide one or more of the agreed
QoS parameters. The Host chooses what action should be done. The Host can
reissue QoS_Setup command to renegotiate the QoS setting for Connection
Handle. The Connection_Handle will be a Connection_Handle for an ACL con-
nection.

Event Parameters:

_Connection_Handle:_   _Size: 2 Bytes (12 Bits meaningful)_

Value	Parameter Description
0xXXXX	Connection Handle that the LM is unable to provide the current QoS requested for.
	Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Bluetooth

5.2.31 Page Scan Mode Change event

Event	Event Code	Event Parameters
Page Scan Mode Change	0x1F	BD_ADDR, Page_Scan_Mode

Description:

The Page Scan Mode Change event indicates that the remote Bluetooth device with the specified BD_ADDR has successfully changed the Page_Scan_Mode.

Event Parameters:

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the remote device.

Page_Scan_Mode: *Size: 1 Byte*

Value	Parameter Description
0x00	Mandatory Page Scan Mode.
0x01	Optional Page Scan Mode I.
0x02	Optional Page Scan Mode II.
0x03	Optional Page Scan Mode III.
0x04 – 0xFF	Reserved.

Bluetooth

5.2.32 Page Scan Repetition Mode Change event

Event	Event Code	Event Parameters
Page Scan Repetition Mode Change	0x20	BD_ADDR, Page_Scan_Repetition_Mode

Description:

The Page Scan Repetition Mode Change event indicates that the remote Bluetooth device with the specified BD_ADDR has successfully changed the Page_Scan_Repetition_Mode (SR).

Event Parameters:

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXX XXXX	BD_ADDR of the remote device.

Page_Scan_Repetition_Mode: *Size: 1 Byte*

Value	Parameter Description
0x00	R0
0x01	R1
0x02	R2
0x03 – 0xFF	Reserved.

6 LIST OF ERROR CODES

6.1 LIST OF ERROR CODES

This section of the document lists the various possible error codes. When a command fails, Error codes are returned to indicate the reason for the error. Error codes have a size of one byte, and the possible range of failure codes is 0x01-0xFF. Section 6.2 provides an error code usage description for each error code.

Error Code	Description
0x01	Unknown HCI Command.
0x02	No Connection.
0x03	Hardware Failure.
0x04	Page Timeout.
0x05	Authentication Failure.
0x06	Key Missing.
0x07	Memory Full.
0x08	Connection Timeout.
0x09	Max Number Of Connections.
0x0A	Max Number Of SCO Connections To A Device.
0x0B	ACL connection already exists.
0x0C	Command Disallowed.
0x0D	Host Rejected due to limited resources.
0x0E	Host Rejected due to security reasons.
0x0F	Host Rejected due to remote device is only a personal device.
0x10	Host Timeout.
0x11	Unsupported Feature or Parameter Value.
0x12	Invalid HCI Command Parameters.
0x13	Other End Terminated Connection: User Ended Connection.
0x14	Other End Terminated Connection: Low Resources.
0x15	Other End Terminated Connection: About to Power Off.
0x16	Connection Terminated by Local Host.
0x17	Repeated Attempts.

Error Code	Description
0x18	Pairing Not Allowed.
0x19	Unknown LMP PDU.
0x1A	Unsupported Remote Feature.
0x1B	SCO Offset Rejected.
0x1C	SCO Interval Rejected.
0x1D	SCO Air Mode Rejected.
0x1E	Invalid LMP Parameters.
0x1F	Unspecified Error.
0x20	Unsupported LMP Parameter Value.
0x21	Role Change Not Allowed
0x22	LMP Response Timeout
0x23	LMP Error Transaction Collision
0x24	LMP PDU Not Allowed
0x25-0xFF	Reserved for Future Use.

Table 6.1: List of Possible Error Codes

Bluetooth

6.2 HCI ERROR CODE USAGE DESCRIPTIONS

The purpose of this section is to give descriptions of how the error codes specified in Table 6.1 on page 745 should be used. It is beyond the scope of this document to give detailed descriptions of all situations where error codes can be used – especially as this may also, in certain cases, be implementation-dependent. However, some error codes that are to be used only in very special cases are described in more detail than other, more general, error codes.

The following error codes are only used in LMP messages, and are therefore not described in this section:

- Unknown LMP PDU (0x19)

- SCO Offset Rejected (0x1B)

- SCO Interval Rejected (0x1C)

- SCO Air Mode Rejected (0x1D)

- Invalid LMP Parameters (0x1E)

Some of the following error code descriptions describe as implementation-dependent whether the error should be returned using a Command Status event or the event associated with the issued command (following a Command Status event with Status=0x00). In these cases, the command can not start executing because of the error, and it is therefore recommended to use the Command Status event. The reason for this suggested course of action is that it is not possible to use the Command Status event in all software architectures.

6.3 UNKNOWN HCI COMMAND (0X01)

The 'Unknown HCI Command' error code is returned by the Host Controller in the Status parameter in a Command Complete event or a Command Status event when the Host Controller receives an HCI Command Packet with an Opcode that it does not recognize. The OpCode given might not correspond to any of the OpCodes specified in this document, or any vendor-specific OpCodes, or the command may not have been implemented. If a Command Complete event is returned, the Status parameter is the only parameter contained in the Return_Parameters event parameter. Which of the two events is used is implementation-dependent.

Bluetooth

6.4 NO CONNECTION (0X02)

The 'No Connection' error code is returned by the Host Controller in the Status parameter in an event when the Host has issued a command which requires an existing connection and there is currently no connection corresponding to the specified Connection Handle or BD Address. If the issued command is a command for which a Command Complete event should be returned, the event containing the error code is a Command Complete event. Otherwise, the event containing the error code is a Command Status event or the event associated with the issued command (following a Command Status event with Status=0x00), depending on the implementation.

6.5 HARDWARE FAILURE (0X03)

The 'Hardware Failure' error code is returned by the Host Controller in the Status parameter in an event when the Host has issued a command and this command can not be executed because of a hardware failure. If the issued command is a command for which a Command Complete event should be returned, the event containing the error code is a Command Complete event. Otherwise, the event containing the error code is a Command Status event or the event associated with the issued command (following a Command Status event with Status=0x00) depending on the implementation.

6.6 PAGE TIMEOUT (0X04)

The 'Page Timeout' error code is returned by the Host Controller in the Status parameter of the Connection Complete event when the Host has issued a Create_Connection command and the specified device to connect to does not respond to a page at baseband level before the page timer expires (a page timeout occurs). The error code can also be returned in the Status parameter of a Remote Name Request Complete event when the Host has issued a Remote_Name_Request command and a temporary connection needs to be established but a page timeout occurs. (The page timeout is set using the Write_Page_Timeout command.)

6.7 AUTHENTICATION FAILED (0X05)

The 'Authentication Failed' error code is returned by the Host Controller in the Status parameter in a Connection Complete event or Authentication Complete event when pairing or authentication fails due to incorrect results in the pairing/ authentication calculations (because of incorrect PIN code or link key).

6.8 KEY MISSING (0X06)

The 'Key Missing' error code is returned by the Host Controller in the Status parameter in a Connection Complete event or Authentication Complete event when pairing fails because of missing PIN code(s).

Bluetooth

6.9   MEMORY FULL (0X07)

The 'Memory Full' error code is returned by the Host Controller in the Status parameter in a Command Complete event when the Host has issued a command that requires the Host Controller to store new parameters and the Host Controller does not have memory capacity for this. This may be the case after the Set_Event_Filter command has been issued. Note that for the Write_Stored_Link_Key command, no error is returned when the Host Controller can not store any more link keys. The Host Controller stores as many link keys as there is free memory to store in, and the Host is notified of how many link keys were successfully stored.

6.10   CONNECTION TIMEOUT (0X08)

Note: this error code is used to indicate a reason for disconnection. It is normally returned in the Reason parameter of a Disconnection Complete event. It is therefore called reason code in the following description.

The 'Connection Timeout' reason code is sent by the Host Controller in an event when the link supervision timer (see) expires and the link therefore is considered to be lost. The link supervision timeout is set using Write_Link_Supervision_Timeout. The event that returns this reason code will most often be a Disconnection Complete event (in the Reason parameter). The event will be returned on both sides of the connection, where one Disconnection Complete event will be sent from the Host Controller to the Host for each Connection Handle that exists for the physical link to the other device.

(It is possible for a link loss to be detected during connection set up, in which case the reason code would be returned in a Connection Complete event.)

6.11   MAX NUMBER OF CONNECTIONS (0X09)

The 'Max Number Of Connections' error code is returned by the Host Controller in the Status parameter of a Command Status event, a Connection Complete event or a Remote Name Request Complete event when the Bluetooth module can not establish any more connections. It is implementation specific whether the error is returned in a Command Status event or the event following the Command Status event (where Status=0x00 in the Command Status event). The reason for this error may be hardware or firmware limitations. Before the error is returned, the Host has issued a Create_Connection, Add_SCO_Connection or Remote_Name_Request command. The error can be returned in a Remote Name Request Complete event when a temporary connection needs to be established to request the name.

Bluetooth

6.12   MAX NUMBER OF SCO CONNECTIONS TO A DEVICE (0X0A)

The 'Max Number Of SCO Connections To A Device' error code is returned by the Host Controller in the Status parameter of a Command Status event or a Connection Complete event (following a Command Status event with Status=0x00) when the maximum number of SCO connections to a device has been reached. Which of the two events that is used depends on the implementation. The device is a device that has been specified in a previously issued Add_SCO_Connection command.

6.13   ACL CONNECTION ALREADY EXISTS (0X0B)

The 'ACL connection already exists' error code is returned by the Host Controller in the Status parameter of a Command Status event or a Connection Complete event (following a Command Status event with Status=0x00) when there already is one ACL connection to a device and the Host tries to establish another one using Create_Connection. Which of the two events that is used depends on the implementation.

6.14   COMMAND DISALLOWED (0X0C)

The 'Command Disallowed' error code is returned by the Host Controller in the Status parameter in a Command Complete event or a Command Status event when the Host Controller is in a state where it is only prepared to accept commands with certain OpCodes and the HCI Command Packet received does not contain any of these OpCodes. The Command Complete event should be used if the issued command is a command for which a Command Complete event should be returned. Otherwise, the Command Status event should be used. The Host Controller is not required to use the 'Unknown HCI Command' error code, since this may require unnecessary processing of the received (and currently not allowed) OpCode. When to use the 'Command Disallowed' error code is mainly implementation-dependent. Certain implementations may, for example, only accept the appropriate HCI response commands after the Connection Request, Link Key Request or PIN Code Request events.
Note: the Reset command should always be allowed.

6.15   HOST REJECTED DUE TO … (0X0D-0X0F)

Note: these error codes are used to indicate a reason for rejecting an incoming connection. They are therefore called reason codes in the following description.

When a Connection Request event has been received by the Host and the Host rejects the incoming connection by issuing the Reject_Connection_ Request command, one of these reason codes is used as value for the Reason parameter. The issued reason code will be returned in the Status parameter of the Connection Complete event that will follow the Command Status event

(with Status=0x00) returned by the Host Controller after the Reject_ Connection_Request command has been issued. The reason code issued in the Reason parameter of the Reject_Connection_Request command will also be sent over the air, so that it is returned in a Connection Complete event on the initiating side. Before this, the initiating side has issued a Create_ Connection command or Add_SCO_Connection command, and has received a Command Status event (with Status=0x00).

6.16   HOST TIMEOUT (0X10)

Note: this error code is used to indicate a reason for rejecting an incoming connection. It is therefore called reason code in the following description.

Assume that a Connection Request event has been received by the Host and that the Host does not issue the Accept_Connection_Request or Reject_ Connection_Request command before the connection accept timer expires (the connection accept timeout is set using Write_Connection_Accept_ Timeout). In this case, the 'Host Timeout' reason code will be sent by the Host Controller in the Status parameter of a Connection Complete event. The reason code will also be sent over the air, so that it is returned in a Connection Complete event on the initiating side. The initiating side has before this issued a Create_Connection or Add_SCO_Connection command and has received a Command Status event (with Status=0x00).

6.17   UNSUPPORTED FEATURE OR PARAMETER VALUE (0X11)

The 'Unsupported Feature or Parameter Value' error code is returned by the Host Controller in the Status parameter in an event when the Host Controller has received a command where one or more parameters have values that are not supported by the hardware (the parameters are, however, within the allowed parameter range specified in this document). If the issued command is a command for which a Command Complete event should be returned, the event containing the error code is a Command Complete event. Otherwise, the event containing the error code is a Command Status event or the event associated with the issued command (following a Command Status event with Status=0x00) depending on the implementation.

6.18   INVALID HCI COMMAND PARAMETERS (0X12)

The 'Invalid HCI Command Parameters' error code is returned by the Host Controller in the Status parameter of an event when the total parameter length (or the value of one or more parameters in a received command) does not conform to what is specified in this document.

The error code can also be returned if a parameter value is currently not allowed although it is inside the allowed range for the parameter. One case is when a command requires a Connection Handle for an ACL connection but the

Host has given a Connection Handle for an SCO connection as a parameter instead. Another case is when a link key, a PIN code or a reply to an incoming connection has been requested by the Host Controller by using an event but the Host replies using a response command with a BD_ADDR for which no request has been made.

If the issued command is a command for which a Command Complete event should be returned, the event containing the error code is a Command Complete event. Otherwise, the event containing the error code is a Command Status event or the event associated with the issued command (following a Command Status event with Status=0x00), depending on the implementation.

6.19 OTHER END TERMINATED CONNECTION: … (0X13-0X15)

Note: these error codes are used to indicate a reason for disconnecting a connection. They are therefore called reason codes in the following description.

When the Host issues the Disconnect command, one of these reason codes is used as value for the reason parameter. The 'Connection Terminated By Local Host' reason code will then be returned in the Reason parameter of the Disconnection Complete event that will follow the Command Status event (with Status=0x00) that is returned by the Host Controller after the Disconnect command has been issued. The reason code issued in the Reason parameter of the Disconnect command will also be sent over the air, so that it is returned in the Reason parameter of a Disconnection Complete event on the remote side.

6.20 CONNECTION TERMINATED BY LOCAL HOST (0X16)

See description in 6.19. This error code is called a reason code, since it is returned in the Reason parameter of a Disconnection Complete event.

6.21 REPEATED ATTEMPTS (0X17)

The 'Repeated Attempts' error code is returned by the Host Controller in the Status parameter in a Connection Complete event or Authentication Complete event when a device does not allow authentication or pairing because too little time has elapsed since an unsuccessful authentication or pairing attempt. See "Link Manager Protocol" on page 185 for a description of how repeated attempts work.

6.22 PAIRING NOT ALLOWED (0X18)

The 'Pairing Not Allowed' error code is returned by the Host Controller in the Status parameter in a Connection Complete event or Authentication Complete event when a device for some reason does not allow pairing. An example may be a PSTN adapter that only allows pairing during a certain time window after a button has been pressed on the adapter.

6.23 UNSUPPORTED REMOTE FEATURE (0X1A)

The 'Unsupported Remote Feature' error code is returned by the Host Controller in the Status parameter of the event associated with the issued command when a remote device that has been specified in the command parameters does not support the feature associated with the issued command. The 'Unsupported Remote Feature' error code can also be used as a value for the Reason parameter in the Disconnect command (as a reason code). The error code will then be sent over the air so that it is returned in the Reason parameter of a Disconnection Complete event on the remote side. In the Disconnection Complete event following a Command Status event (where Status=0x00) on the local side on which the Disconnect command has been issued, the Reason parameter will however contain the reason code 'Connection Terminated By Local Host'. (The 'Unsupported Remote Feature' error code is called 'Unsupported LMP Feature' in the LMP specification, see .)

6.24 UNSPECIFIED ERROR (0X1F)

The 'Unspecified error' error code is used when no other error code specified in this document is appropriate to use.

6.25 UNSUPPORTED LMP PARAMETER VALUE (0X20)

The 'Unsupported LMP Parameter Value' error code is returned by the Host Controller in the Status parameter of the event associated with the issued command when a remote device that has been specified in the command parameters sent back an LMP message containing the LMP error code 0x20, 'Unsupported parameter values' (see).

6.26 ROLE CHANGE NOT ALLOWED (0X21)

The 'Role Change Not Allowed' error code is returned by the Host Controller in the Status parameter in a Connection Complete event or Role Change event when role change is not allowed. If the local Host issues the Switch_Role command and the remote device rejects the role change, the error code will be returned in a Role Change event. If a connection fails because a device accepts an incoming ACL connection with a request for role change and the role change is rejected by the initiating device, the error code will be returned in a Connection Complete event on both sides.

6.27 LMP RESPONSE TIMEOUT (0X22)

The 'LMP Response Timeout' error code is returned by the Host Controller in the Status parameter in a Command Complete event or an event associated with the issued command following a Command Status event with Status=0x00, when the remote device does not respond to the LMP PDUs from

Bluetooth

the local device as a result of the issued command within LMP response time-out. (See "Link Manager Protocol" on page 185)

6.28   LMP ERROR TRANSACTION COLLISION (0X23)

The 'LMP Error Transaction Collision' error code is returned by the Host Controller in the Status parameter of the event associated with the issued command when a remote device that has been specified in the command parameters sends back an LMP message containing the LMP error code 0x23, "LMP Error Transaction Collision" (see "Link Manager Protocol" on page 185).

6.29   LMP PDU NOT ALLOWED (0X24)

The 'LMP PDU Not Allowed' error code is returned by the Host Controller in the Status parameter of the event associated with the issued command when a remote device that has been specified in the command parameters sends back an LMP message containing the LMP error code 0x24, "PDU Not Allowed" (see "Link Manager Protocol" on page 185).

Host Controller Interface Functional Specification

Bluetooth

7 LIST OF ACRONYMS AND ABBREVIATIONS

Acronym or abbreviation	Complete name
ACL	Asynchronous Connection Less
BD_ADDR	Bluetooth Device Address
DH	Data High rate
DIAC	Dedicated Inquiry Access Code
DM	Data Medium rate
DUT	Device Under Test
DV	Data Voice
GIAC	General Inquiry Access Code
HCI	Host Controller Interface
L2CAP	Logical Link Control and Adaptation Protocol
L_CH	Logical Channel
LAP	Lower Address Part
LC	Link Controller
LM	Link Manager
LMP	Link Manager Protocol
OCF	Opcode Command Field
OGF	OpCode Group Field
RF	Radio Frequency
RSSI	Received Signal Strength Indication
SCO	Synchronous Connection Oriented
TBD	To Be Defined
UA	User Asynchronous
UI	User Isochronous
US	User Synchronous
USB	Universal Serial Bus

Table 7.1: List of Acronyms and Abbreviations

Bluetooth

8 LIST OF FIGURES

Bluetooth.

9 LIST OF TABLES

Bluetooth