11.2.7.18 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.
0x04–0xFF	Reserved.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter description
0x00	Write_Scan_Enable command succeeded.
0x01–0xFF	Write_Scan_Enable command failed. See Table 109 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.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 366 of 1169

11.2.7.19 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 parameters. 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 that Page Scan is only performed when Page_Scan is enabled (see 11.2.7.17 and 11.2.7.18).

A changed Page_Scan_Interval could change the local Page_Scan_Repetition_Mode (see Clause 8, 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 109 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 ms Range: 11.25 ms– 2560 ms

Copyright © 2002 IEEE. All rights reserved.

Page_Scan_Window: *Size: 2 Bytes*

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

Event(s) generated (unless masked away):

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

11.2.7.20 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 that Page Scan is only performed when Page_Scan is enabled (see 11.2.7.17 and 11.2.7.18). A changed Page_Scan_Interval could change the local Page_Scan_Repetition_Mode (see Clause 8, 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 ms Range: 11.25 ms–2560 ms Default: *N* = 0x0800 Time = 1.28 s

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 368 of 1169

Page_Scan_Window: *Size: 2 Bytes*

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

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 109 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.

11.2.7.21 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 that Inquiry Scan is only performed when Inquiry_Scan is enabled (see 11.2.7.17 and 11.2.7.18).

Command Parameters:

None.

Copyright © 2002 IEEE. All rights reserved.

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 109 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 ms Range: 11.25–2560 ms

Inquiry_Scan_Window: *Size: 2 Bytes*

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

Event(s) generated (unless masked away):

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

11.2.7.22 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 that Inquiry Scan is only performed when Inquiry_Scan is enabled (see 11.2.7.17 and 11.2.7.18).

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 370 of 1169

Command Parameters:

Inquiry_Scan_Interval: *Size: 2 Bytes*

Value	Parameter description
N = 0xXXXX	Size: 2 Bytes Range: 0x0012–0x1000 Time = *N* * 0.625 ms Range: 11.25–2560 ms Default: *N* = 0x0800 Time = 1.28 s

Inquiry_Scan_Window: *Size: 2 Bytes*

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

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 109 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.

11.2.7.23 Read_Authentication_Enable

Command	OCF	Command parameters	Return parameters
HCI_Read_ Authentication_Enable	0x001F	—	Status, Authentication_Enable

Copyright © 2002 IEEE. All rights reserved.

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 that 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 109 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.

11.2.7.24 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.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 372 of 1169

Note that 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 109 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.

11.2.7.25 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 that changing this parameter does not affect existing connections.

Command Parameters:

None.

Copyright © 2002 IEEE. All rights reserved.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter description
0x00	Read_Encryption_Mode command succeeded.
0x01–0xFF	Read_Encryption_Mode command failed. See Table 109 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.

11.2.7.26 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 that changing this parameter does not affect existing connections. A temporary link key shall be used when both broadcast and point-to-point traffic shall be encrypted.

The Host shall 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 shall 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 shall 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

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 374 of 1169

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.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter description
0x00	Write_Encryption_Mode command succeeded.
0x01–0xFF	Write_Encryption_Mode command failed. See Table 109 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.

11.2.7.27 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.

Copyright © 2002 IEEE. All rights reserved.

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 109 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.

11.2.7.28 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 109 for list of Error Codes.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 376 of 1169

Event(s) generated (unless masked away):

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

11.2.7.29 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 109 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).

Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 377 of 1169

Value	Parameter description
XXXXXnnnXX	Linear_PCM_Bit_Pos: number of 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.

11.2.7.30 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).

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 378 of 1169

Value	Parameter description
XXXX1XXXXX	Input Sample Size: 16 bit (only for Liner PCM).
XXXXXnnnXX	Linear_PCM_Bit_Pos: number of 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.
00011000XX	Default Condition. (X means that there is no default value for the corresponding bit. The manufacturer may use any value.)

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter description
0x00	Write_Voice_Setting command succeeded.
0x01–0xFF	Write_Voice_Setting command failed. See Table 109 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.

11.2.7.31 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

Copyright © 2002 IEEE. All rights reserved.

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 109 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 ms 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.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 380 of 1169

11.2.7.32 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 ms Size: 11 bits Range: 0x0001–0x07FF

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 109 for list of Error Codes.

Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 381 of 1169

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.

11.2.7.33 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 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 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 109 for list of Error Codes.

Num_Broadcast_Retran: *Size: 1 Byte*

Value	Parameter description
N = 0xXX	Number of Broadcast Retransmissions = *N* Range 0x00–0xFF

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 382 of 1169

Event(s) generated (unless masked away):

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

11.2.7.34 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 changes.

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 109 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.

Copyright © 2002 IEEE. All rights reserved.

11.2.7.35 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 109 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.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 384 of 1169

11.2.7.36 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 109 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.

Copyright © 2002 IEEE. All rights reserved.

11.2.7.37 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 shall 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 109 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).

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 386 of 1169

Transmit_Power_Level: *Size: 1 Byte*

Value	Parameter description
N = 0xXX	Size: 1 Byte (signed integer) Range: -30 ≤ *N* ≤ 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.

11.2.7.38 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 disable SCO flow control.

Note that the SCO_Flow_Control_Enable setting can only be changed if no connections 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 109 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.

Copyright © 2002 IEEE. All rights reserved.

Event(s) generated (unless masked away):

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

11.2.7.39 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 that 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 109 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.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 388 of 1169

11.2.7.40 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 for data and/or voice sent 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. If flow control is turned on for HCI ACL Data Packets and off for HCI SCO Data Packets, Host_Number_Of_Completed_Packets commands sent by the Host should only contain Connection Handles for ACL connections. If flow control is turned off for HCI ACL Data Packets and on for HCI SCO Data Packets, Host_Number_Of_Completed_Packets commands sent by the Host should only contain Connection Handles for SCO connections. If flow control is turned on for HCI ACL Data Packets and HCI SCO Data Packets, the Host will send Host_Number_Of_Completed_Packets commands both for ACL connections and SCO connections. Note that the Flow_Control_Enable setting must only be changed if no connections exist.

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** for HCI ACL Data Packets and **off** for HCI SCO Data Packets in direction from Host Controller to Host.
0x02	Flow control **off** for HCI ACL Data Packets and **on** for HCI SCO Data Packets in direction from Host Controller to Host.
0x03	Flow control **on** both for HCI ACL Data Packets and HCI SCO Data Packets in direction from Host Controller to Host.
0x04–0xFF	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 109 for list of Error Codes.

Copyright © 2002 IEEE. All rights reserved.

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.

11.2.7.41 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 shall 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 that 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.

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.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 390 of 1169

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 109 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.

11.2.7.42 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_Packets 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

Copyright © 2002 IEEE. All rights reserved.

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 that 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.

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

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 392 of 1169

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.

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

Note that 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 s. 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 109 for list of Error Codes.

Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 393 of 1169

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 ms (1 Baseband slot) Range for *N*: 0x0001–0xFFFF Time Range: 0.625ms–40.9 s

Event(s) generated (unless masked away):

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

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

Note that 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 s. By using the No Link_Supervision_Timeout setting, the scalability of the Park mode is not limited.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 394 of 1169

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 ms (1 Baseband slot) Range for *N*: 0x0001–0xFFFF Time Range: 0.625 ms–40.9 s **Default**: *N* = 0x7D00 Link_Supervision_Timeout = 20 s

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 109 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.

Copyright © 2002 IEEE. All rights reserved.

11.2.7.45 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 IACs 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). 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 109 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.

11.2.7.46 Read_Current_IAC_LAP

Command	OCF	Command parameters	Return parameters
HCI_Read_Current_IAC_LAP	0x0039	—	Status, Num_Current_IAC, IAC_LAP[i]

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 396 of 1169

Description:

This command reads the LAP(s) used to create the IAC(s) that the local Bluetooth device is simultaneously scanning for during Inquiry Scans. All Bluetooth devices are required to support at least one IAC, the GIAC. 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 109 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.

11.2.7.47 Write_Current_IAC_LAP

Command	OCF	Command Parameters	Return parameters
HCI_Write_Current_IAC_LAP	0x003A	Num_Current_IAC, IAC_LAP[i]	Status

Copyright © 2002 IEEE. All rights reserved.

Description:

This command writes the LAP(s) used to create the IAC that the local Bluetooth device is simultaneously scanning for during Inquiry Scans. All Bluetooth devices are required to support at least one IAC, the GIAC. Some Bluetooth devices support additional IACs.

Note that this command writes over the current IACs used by the Bluetooth device. If the value of the Num_Current_IAC is more than the number of supported IACs, then only the first, X IAC (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.

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 109 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.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 398 of 1169

11.2.7.48 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 that the timer T_mandatory_pscan will be reset at each new inquiry response. For details, see Clause 8 (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 109 for list of Error Codes.

Page_Scan_Period_Mode: *Size: 1 Byte*

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

Event(s) generated (unless masked away):

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

Copyright © 2002 IEEE. All rights reserved.

11.2.7.49 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 that the timer T_mandatory_pscan will be reset at each new inquiry response. For details, see Clause 8 (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 109 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.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 400 of 1169

11.2.7.50 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 shall be applied. For details, see Clause 8 (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 109 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.

Copyright © 2002 IEEE. All rights reserved.

11.2.7.51 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 shall be applied. For details see Clause 8 (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 109 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.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 402 of 1169

11.2.8 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.

11.2.8.1 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 Bluetooth SIG, Inc.

Copyright © 2002 IEEE. All rights reserved.

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 109 for list of Error Codes.

HCI_Version: *Size: 1 Byte*

Value	Parameter description
0x00	Bluetooth HCI Specification 1.0B.
0x01	Bluetooth HCI Specification 1.1.
0x02–0xfFF	Reserved for future use.

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 68 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 68 in the Link Manager Protocol for assigned values (CompId).

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 404 of 1169

LMP_Subversion: *Size: 2 Bytes*

Value	Parameter description
0xXXXX	Subversion of the Current LMP in the Bluetooth Hardware, see Table 68 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.

11.2.8.2 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 Clause 9.

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 109 for list of Error Codes.

LMP_Features: *Size: 8 Bytes*

Value	Parameter description
0xXXXXXXXX XXXXXXXX	Bit Mask List of LMP features. For details, see Clause 9.

Event(s) generated (unless masked away):

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

Copyright © 2002 IEEE. All rights reserved.

11.2.8.3 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 that 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 109 for list of Error Codes.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 406 of 1169

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.

11.2.8.4 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.

Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 407 of 1169

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter description
0x00	Read_Country_Code command succeeded.
0x01–0xFF	Read_Country_Code command failed. See Table 109 for list of Error Codes.

Country_Code: *Size: 1 Byte*

Value	Parameter description
0x00	North America, Europe (except France), and Japan
0x01	France
0x04–FF	Reserved for future use.

Event(s) generated (unless masked away):

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

11.2.8.5 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 Clause 8 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 109 for list of Error Codes.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 408 of 1169

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.

11.2.9 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 did not 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 did not 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.

Copyright © 2002 IEEE. All rights reserved.

11.2.9.1 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 shall 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 did not 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 > 0 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 109 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 read. Range: 0x0000–0x0EFF (0x0F00–0x0FFF Reserved for future use).

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 410 of 1169

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.

11.2.9.2 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 shall 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 did not 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 > 0 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).

Copyright © 2002 IEEE. All rights reserved.

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 109 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.

11.2.9.3 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 shall 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).

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 412 of 1169

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter description
0x00	Get_Link_Quality command succeeded.
0x01–0xFF	Get_Link_Quality command failed. See Table 109 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).

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.

11.2.9.4 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 RSSI and the limits of the Golden Receive Power Range (see 7.4.7) for a connection handle to another Bluetooth device. The Connection_Handle shall 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.

Copyright © 2002 IEEE. All rights reserved.

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 109 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).

RSSI: *Size: 1 Byte*

Value	Parameter description
N = 0xXX	Size: 1 Byte (signed integer) Range: -128 ≤ *N* ≤ 127 Units: dB

Event(s) generated (unless masked away):

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

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 414 of 1169

11.2.10 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.

11.2.10.1 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 Nontesting Mode operation, the Loopback Mode is set to Nontesting 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 Figure 138.

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

Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 415 of 1169

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 Figure 139, where the rightmost device is set to Remote Loopback Mode and the leftmost device is set to Nontesting Mode. This allows the Bluetooth Air link to be tested without any other variables.

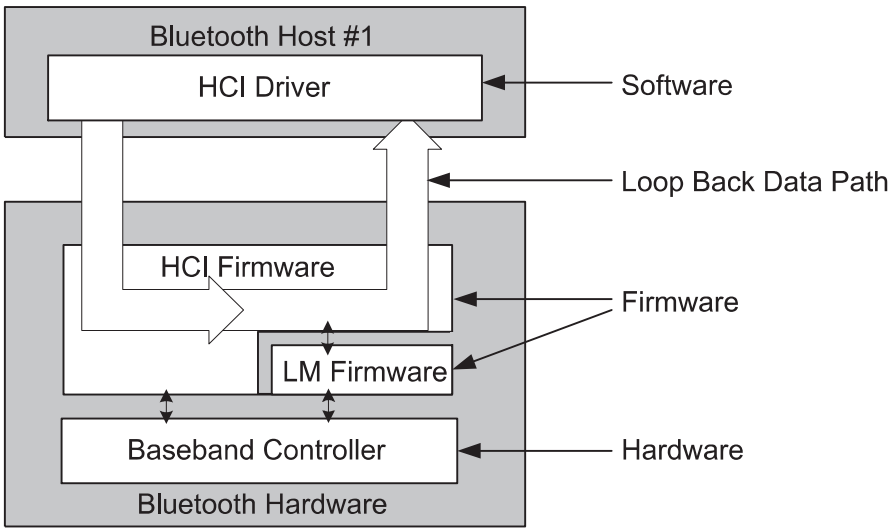


Figure 138—Local loopback mode

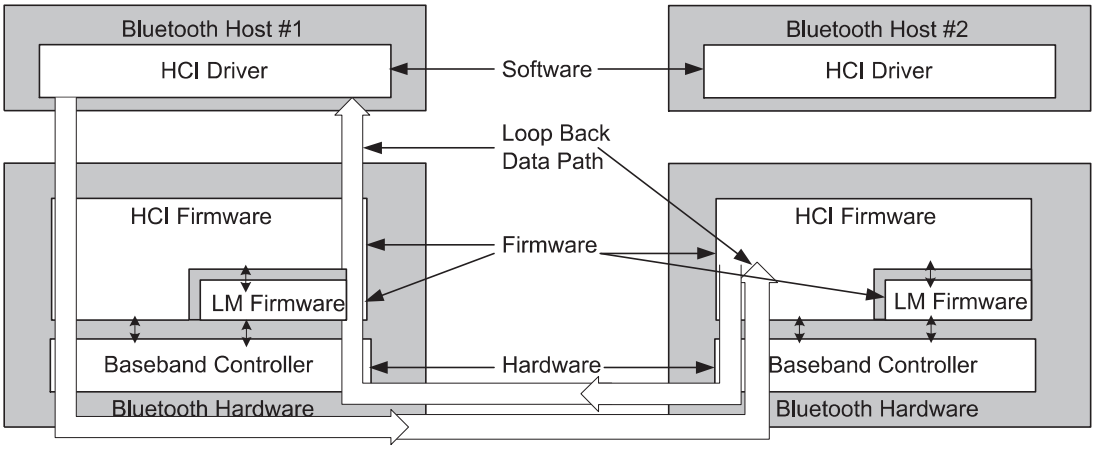


Figure 139—Remote loopback mode

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 416 of 1169

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter description
0x00	Read_Loopback_Mode command succeeded.
0x01–0xFF	Read_Loopback_Mode command failed. See Table 109 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.

11.2.10.2 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 Nontesting Mode operation, the Loopback Mode is set to Nontesting 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 Figure 140.

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.

Copyright © 2002 IEEE. All rights reserved.

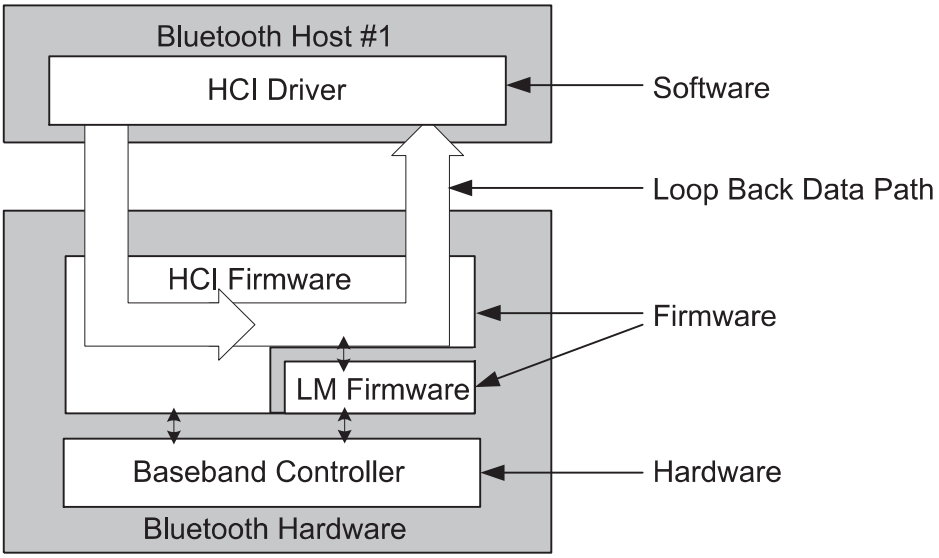


Figure 140—Local loopback mode

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 Figure 141, where the rightmost device is set to Remote Loopback Mode and the leftmost device is set to Nontesting Mode. This allows the Bluetooth Air link to be tested without any other variables.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 418 of 1169

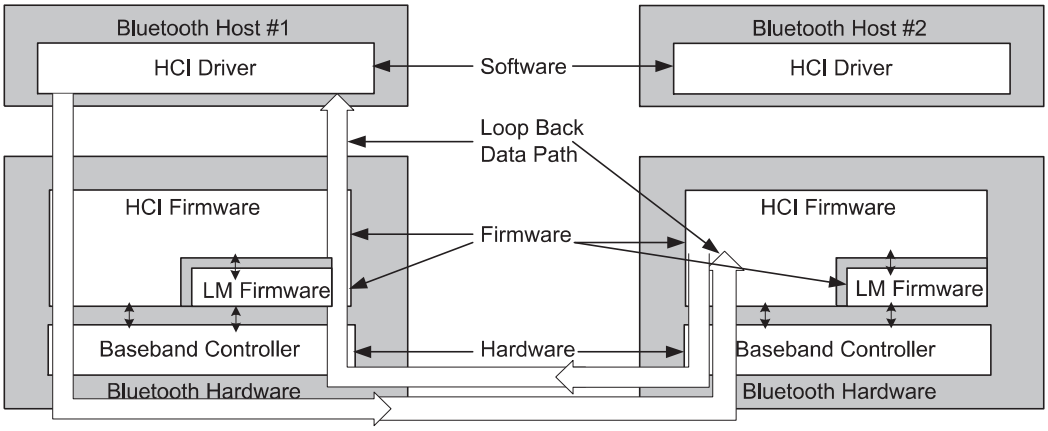


Figure 141—Remote loopback mode

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 109 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.

Copyright © 2002 IEEE. All rights reserved.

11.2.10.3 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 Clause 9. The Host issues this command when it wants the local device to be the DUT for the Testing scenarios as described in Annex E. 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 109 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.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 420 of 1169

11.3 Events

11.3.1 Event

In addition to the events listed in Table 108, 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.

Table 108—List of supported events

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.

Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 421 of 1169

Table 108—List of supported events *(continued)*

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.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 422 of 1169

Table 108—List of supported events *(continued)*

Event	Event summary description
Loopback command event	The Loopback Command event is used to loop back most commands that the Host sends to the Host Controller.
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).

11.3.2 Possible events

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

11.3.2.1 Inquiry complete event

Event	Event code	Event parameters
Inquiry Complete	0x01	Status

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.

Copyright © 2002 IEEE. All rights reserved.

Event Parameters:

Status: *Size: 1 Byte*

Value	Parameter description
0x00	Inquiry command completed successfully.
0x01–0xFF	Inquiry command failed. See Table 109 for list of Error Codes.

11.3.2.2 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 that responded.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 424 of 1169

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

Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 425 of 1169

11.3.2.3 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 that 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 109 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.

Link_Type: *Size: 1 Byte*

Value	Parameter description
0x00	SCO connection (Voice Channels).
0x01	ACL connection (Data Channels).
0x02–0xFF	Reserved for future use.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 426 of 1169

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.

11.3.2.4 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, that requests 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.

Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 427 of 1169

11.3.2.5 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 that 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 109 for list of Error Codes.

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

Value	Parameter description
0xXXXX	Connection Handle that was disconnected. Range: 0x0000–0x0EFF (0x0F00–0x0FFF Reserved for future use)

Reason: *Size: 1 Byte*

Value	Parameter description
0xXX	Reason for disconnection. See Table 109 for list of Error Codes.

11.3.2.6 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 shall be a Connection_Handle for an ACL connection.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 428 of 1169

Event Parameters:

Status: *Size: 1 Byte*

Value	Parameter description
0x00	Authentication Request successfully completed.
0x01–0xFF	Authentication Request failed to complete. See Table 109 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).

11.3.2.7 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 that the Remote_Name Parameter is a string parameter. Endianess does therefore not apply to the Remote_Name Parameter. The first byte of the name is received first.

Event Parameters:

Status: *Size: 1 Byte*

Value	Parameter description
0x00	Remote_Name_Request command succeeded.
0x01–0xFF	Remote_Name_Request command failed. See Table 109 for list of Error Codes.

BD_ADDR: *Size: 6 Bytes*

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

Copyright © 2002 IEEE. All rights reserved.

Remote_Name: *Size: 248 Bytes*

Value	Parameter description
Name[248]	A UTF-8 encoded user-friendly descriptive name for the remote device. If the name contained in the parameter is shorter than 248 bytes, the end of the name is indicated by a NULL byte (0x00), and the following bytes (to fill up 248 bytes, which is the length of the parameter) do not have valid values.

11.3.2.8 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 109 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.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 430 of 1169

11.3.2.9 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 that 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 109 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).

11.3.2.10 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 semipermanent 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 semipermanent link keys) is now being used in the piconet.

Note that for a master, the change from a semipermanent 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

Copyright © 2002 IEEE. All rights reserved.

handle. A temporary link key shall 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 109 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 Semipermanent Link Key.
0x01	Using Temporary Link Key.

11.3.2.11 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 Clause 9.

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 109 for list of Error Codes.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 432 of 1169

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

Value	Parameter description
0xXXXX	Connection Handle that 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 Clause 9.

11.3.2.12 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 Bluetooth SIG, Inc. 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 109 for list of Error Codes.

Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 433 of 1169

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

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

LMP_Version: *Size: 1 Byte*

Value	Parameter description
0xXX	Version of the Current LMP in the remote Bluetooth Hardware, see Table 68 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 68 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 68 in the Link Manager Protocol for assigned values (SubVersNr).

11.3.2.13 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 Clause 9.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 434 of 1169

Event Parameters:

Status: *Size: 1 Byte*

Value	Parameter description
0x00	QoS_Setup command succeeded.
0x01–0xFF	QoS_Setup command failed. See Table 109 for list of Error Codes.

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

Value	Parameter description
0xXXXX	Connection Handle that 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.

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.

Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 435 of 1169

Delay_Variation: *Size: 4 Bytes*

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

11.3.2.14 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
0xXX	The Number of HCI command packets that 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.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 436 of 1169

11.3.2.15 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 109 for list of Error Codes.

Num_HCI_Command_Packets: *Size: 1 Byte*

Value	Parameter description
0xXX	The Number of HCI command packets that 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 that caused this event and is pending completion.

Copyright © 2002 IEEE. All rights reserved.

11.3.2.16 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
0xXX	These Hardware_Codes will be implementation-specific, and will be assigned to indicate various hardware problems.

11.3.2.17 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 shall 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).

11.3.2.18 Role Change event

Event	Event code	Event parameters
Role Change	0x12	Status, BD_ADDR, New_Role

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 438 of 1169

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 109 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.

11.3.2.19 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.

Copyright © 2002 IEEE. All rights reserved.

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 in 11.2.7.38 and 11.2.7.39.)

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).

HC_Num_Of_Completed_Packets [i]: *Size: Number_of_Handles * 2 Bytes*

Value	Parameter description
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

11.3.2.20 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.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 440 of 1169

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 109 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).

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.

Copyright © 2002 IEEE. All rights reserved.

Interval: *Size: 2 Bytes*

Value	Parameter description
0xXXXX	Hold: Number of Baseband slots to wait in Hold Mode. Hold Interval = *N* * 0.625 ms (1 Baseband slot) Range for *N*: 0x0000–0xFFFF Time Range: 0-40.9 s 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 s Park: Number of Baseband slots between consecutive beacons. Interval Length = *N* * 0.625 ms (1 Baseband slot) Range for *N*: 0x0000–0xFFFF Time Range: 0-40.9 s

11.3.2.21 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: 0x01–0x0B

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 442 of 1169

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.

11.3.2.22 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 shall 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 that 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 shall 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 Clause 9.)

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.

11.3.2.23 Link key request event

Event	Event code	Event parameters
Link Key Request	0x17	BD_ADDR

Copyright © 2002 IEEE. All rights reserved.

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 that 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 shall 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 Clause 9.)

Event Parameters:

BD_ADDR: *Size: 6 Bytes*

Value	Parameter description
0xXXXXXXXXXXXX	BD_ADDR of the Device for which a stored link key is being requested.

11.3.2.24 Link key notification event

Event	Event code	Event parameters
Link Key Notification	0x18	BD_ADDR, Link_Key,Key_Type

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 decide to store the Link Key in the Host Controller's Link Key Storage by using the Write_Stored_Link_Key command. The Key_Type event parameter informs the Host about which key type (combination key, local unit key or remote unit key) that has been used during pairing. If pairing with unit key is not supported, the Host can, for instance, discard the key or disconnect the link.

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.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 444 of 1169

Link_Key: *Size: 16 Bytes*

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

Key_Type: *Size: 1 Bytes*

Value	Parameter description
0x00	Combination Key.
0x01	Local Unit Key.
0x02	Remote Unit Key.
0x03–0xFF	Reserved.

11.3.2.25 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 11.2.10.1 for a description of which commands that are not looped back. The HCI_Command_Packet event parameter contains the entire HCI Command Packet including the header.

Note that 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.

11.3.2.26 Data buffer overflow event

Event	Event code	Event parameters
Data Buffer Overflow	0x1A	Link_Type

Copyright © 2002 IEEE. All rights reserved.

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.

11.3.2.27 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 maximum allowed length, in number of slots, for baseband packets transmitted by the local device, changes. The Connection_Handle will be a Connection_Handle for an ACL connection.

Event Parameters:

Connection_Handle: *Size: 2 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
0x01, 0x03, 0x05	Maximum number of slots allowed to use for baseband packets, see 9.3.22 and 9.5.1.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 446 of 1169

11.3.2.28 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 109 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.

11.3.2.29 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

Copyright © 2002 IEEE. All rights reserved.

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 109 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.
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

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 448 of 1169

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.

11.3.2.30 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 provide the current QoS requirement for the Connection Handle. This event indicates 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 connection.

Copyright © 2002 IEEE. All rights reserved.

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).

11.3.2.31 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.

11.3.2.32 Page scan repetition mode change event

Event	Event code	Event parameters
Page Scan Repetition Mode Change	0x20	BD_ADDR, Page_Scan_Repetition_Mode

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 450 of 1169

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
0xXXXXXXXXXXXX	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.

Copyright © 2002 IEEE. All rights reserved.

11.4 List of error codes

11.4.1 List of error codes

Table 109 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. Subclause 11.4.2 provides an error code usage description for each error code.

Table 109—List of possible error codes

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.
0x18	Pairing Not Allowed.
0x19	Unknown LMP PDU.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 452 of 1169

Table 109—List of possible error codes *(continued)*

Error Code	Description
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	Encryption Mode Not Acceptable
0x26	Unit Key Used
0x27	QoS is Not Supported
0x28	Instant Passed
0x29	Pairing with Unit Key Not Supported
0x2A-0xFF	Reserved for future use.

11.4.2 HCI error code usage descriptions

The purpose of this subclause is to give descriptions of how the error codes specified in 11.4.1 should be used. It is beyond the scope of this standard 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 subclause:

— 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

Copyright © 2002 IEEE. All rights reserved.

(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.

11.4.3 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 standard, 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.

11.4.4 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 that 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.

11.4.5 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 cannot 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.

11.4.6 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.)

11.4.7 Authentication Failure (0x05)

The “Authentication Failure” 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 an incorrect PIN code or link key).

"The “Authentication Failure” 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

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 454 of 1169

Disconnect command has been issued, the Reason parameter will, however, contain the reason code "Connection Terminated By Local Host".

11.4.8 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).

11.4.9 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 cannot 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.

11.4.10 Connection Timeout (0x08)

Note that 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 Annex F) 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 setup, in which case, the reason code would be returned in a Connection Complete event.)

11.4.11 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 cannot 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.

11.4.12 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.

Copyright © 2002 IEEE. All rights reserved.

11.4.13 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.

11.4.14 Command Disallowed (0x0C)

The “Command Disallowed” error code is returned by the Host Controller in the Status parameter in a ComNSmand 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 that the Reset command should always be allowed.

11.4.15 Host Rejected due to … (0x0D–0x0F)

Note that 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).

11.4.16 Host Timeout (0x10)

Note that 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).

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 456 of 1169

11.4.17 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 standard). 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.

11.4.18 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 standard.

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 such 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.

11.4.19 Other End Terminated Connection: … (0x13-0x15)

Note that 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.

11.4.20 Connection Terminated By Local Host (0x16)

This error code is called a reason code, since it is returned in the Reason parameter of a Disconnection Complete event. See description in 11.4.19.

11.4.21 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 Clause 9 for a description of how repeated attempts work.

Copyright © 2002 IEEE. All rights reserved.

11.4.22 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.

11.4.23 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 Clause 9.)

11.4.24 Unspecified error (0x1F)

The “Unspecified error” error code is used when no other error code specified in this document is appropriate to use.

11.4.25 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 Clause 9).

11.4.26 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.

11.4.27 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 the local device as a result of the issued command within LMP response timeout (see Clause 9).

11.4.28 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 Clause 9).

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 458 of 1169

11.4.29 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 Clause 9).

11.4.30 Encryption Mode Not Acceptable (0X25)

The “Encryption Mode Not Acceptable” error code is returned by the Host Controller in the Status parameter of a Connection Complete event or an Encryption Change event when no agreement can be reached on which encryption mode to use.

11.4.31 Unit Key Used (0X26)

The “Unit Key Used” error code is returned by the Host Controller in the Status parameter of a Command Status event or a Change Connection Link Key Complete event (following a Command Status event with Status = 0x00) if the link key cannot be changed because it is a unit key.

11.4.32 QoS Is Not Supported (0X27)

The “QoS Is Not Supported” error code is returned by the Host Controller in the Status parameter of a Command Status event or a QoS Setup Complete event (following a Command Status event with Status = 0x00) if the requested QoS is not supported.

11.4.33 Instant Passed (0X28)

The “Instant Passed” error code is returned by the Host Controller in the Status parameter of a Role Change event if a role switch cannot be performed because the instant at which the role switch shall start is in the past.

11.4.34 Pairing With Unit Key Not Supported (0X29)

Note that this error code can be used to indicate a reason for disconnecting a connection. It is, therefore, called reason code in the following description.

When the Host issues the Disconnect command, this reason code can be used as a value for the Reason parameter (if the Key_Type parameter in a Link Key Notification event indicates that unit key is used and this is not supported). The 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.

Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 459 of 1169

12. Service access point interfaces and primitives

12.1 IEEE 802 interfaces

Figure 142 indicates the relationship of the Bluetooth protocol stack to this clause. This clause describes the functions, features, protocol, services, and SAP interfaces between the MAC and LLC sublayers within the data link layer of the ISO/IEC 8802 LAN Protocol (IEEE 802.2). The LLC sublayer constitutes the top sublayer in the data link layer (see Figure 143) and is common to the various medium access methods that are defined and supported by the ISO/IEC 8802 activity.

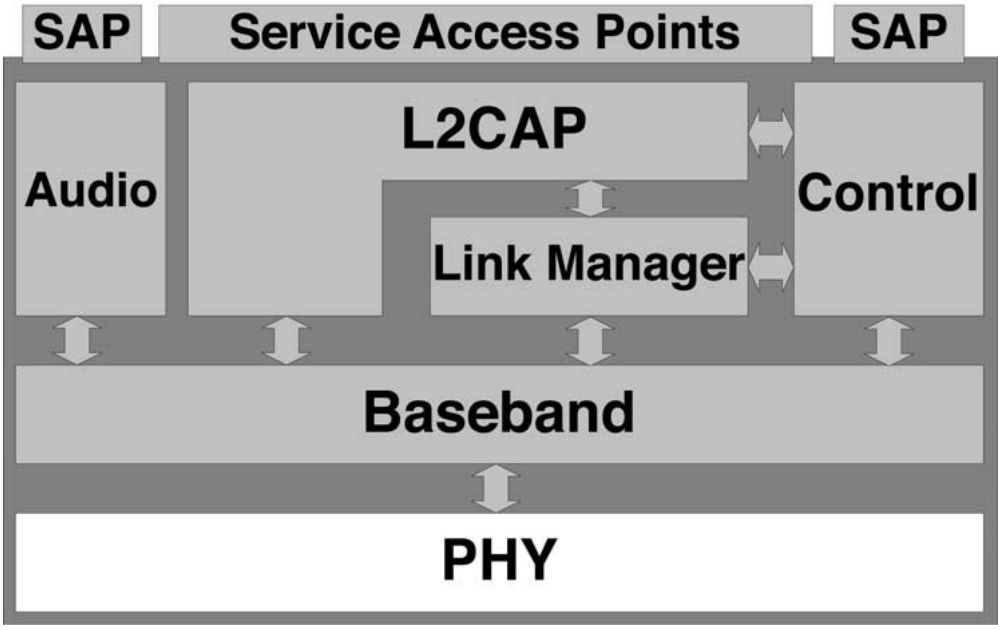


Figure 142—SAP relationships

Copyright © 2002 IEEE. All rights reserved.