Supplement to the Bluetooth Core Specification

Specification of the Bluetooth® System

CSS Version 7

Bluetooth®

Publication Date: Dec 06 2016

TABLE OF CONTENTS

Part A
DATA TYPES SPECIFICATION

Part B
COMMON PROFILE AND SERVICE ERROR CODES

Part C
SERVICES PERMITTED TO USE SECURITY MODE 4 LEVEL 0

Supplement to Bluetooth Core Specification

Part A

DATA TYPES SPECIFICATION

CONTENTS

1 DATA TYPES DEFINITIONS AND FORMATS

This part defines the basic data types used for Extended Inquiry Response (EIR), Advertising Data (AD), Scan Response Data (SRD), Additional Controller Advertising Data (ACAD), and OOB data blocks. Additional data types may be defined in profile specifications.

Each data type shall only be used in accordance with the requirements specified in Table 1.1.

Data type	**Context**				
	EIR	**AD**	**SRD**	**ACAD**	**OOB**
Service UUID	O	O	O	O	O
Local Name	C1	C1	C1	X	C1
Flags	C1	C1	X	X	C1
Manufacturer Specific Data	O	O	O	O	O
TX Power Level	O	O	O	X	O
Secure Simple Pairing OOB	X	X	X	X	O
Security Manager OOB	X	X	X	X	O
Security Manager TK Value	X	X	X	X	O
Slave Connection Interval Range	X	O	O	X	O
Service Solicitation	X	O	O	X	O
Service Data	X	O	O	O	O
Appearance	X	C2	C2	X	C1
Public Target Address	X	C2	C2	X	C1
Random Target Address	X	C2	C2	X	C1
Advertising Interval	X	C1	C1	X	C1
LE Bluetooth Device Address	X	X	X	X	C1
LE Role	X	X	X	X	C1
Uniform Resource Identifier	O	O	O	X	O
LE Supported Features	X	C1	C1	X	C1
Channel Map Update Indication	X	X	X	C1	X

Table 1.1: Permitted usages for data types

O: Optional in this context (may appear more than once in a block).

C1: Optional in this context; shall not appear more than once in a block.

C2: Optional in this context; shall not appear more than once in a block and shall not appear in both the AD and SRD of the same extended advertising interval.

X: Reserved for future use.

The values for the data types are listed in the Bluetooth Assigned Numbers document.

All numerical multi-byte entities and values associated with the following data types shall use little-endian byte order.

1.1 SERVICE UUID

GAP and GATT service UUIDs should not be included in a Service UUIDs AD type, for either a complete or incomplete list.

1.1.1 Description

The Service UUID data type is used to include a list of Service or Service Class UUIDs.

There are six data types defined for the three sizes of Service UUIDs that may be returned:

- 16-bit Bluetooth Service UUIDs
- 32-bit Bluetooth Service UUIDs
- Global 128-bit Service UUIDs

Two Service UUID data types are assigned to each size of Service UUID. One Service UUID data type indicates that the Service UUID list is incomplete and the other indicates the Service UUID list is complete.

A packet or data block shall not contain more than one instance for each Service UUID data size. If a device has no Service UUIDs of a certain size, 16-, 32-, or 128-bit, the corresponding field in the extended inquiry response or advertising data packet shall be marked as complete with no Service UUIDs. An omitted Service UUID data type shall be interpreted as an empty incomplete-list.

16-bit and 32-bit UUIDs shall only be used if they are assigned by the Bluetooth SIG. The Bluetooth SIG may assign 16-bit and 32-bit UUIDs to member companies or organizations.

1.1.2 Format

Data Type	Description
<<Incomplete List of 16-bit Service UUIDs>>	More 16-bit Service UUIDs available
<<Complete List of 16-bit Service UUIDs>>	Complete list of 16-bit Service UUIDs
<<Incomplete List of 32-bit Service UUIDs>>	More 32-bit Service UUIDs available
<<Complete List of 32-bit Service UUIDs>>	Complete list of 32-bit Service UUIDs
<<Incomplete List of 128-bit Service UUIDs>>	More 128-bit Service UUIDs available
<<Complete List of 128-bit Service UUIDs>>	Complete list of 128-bit Service UUIDs

Table 1.2: Service UUID Data Types

1.2 LOCAL NAME

1.2.1 Description

The Local Name data type shall be the same as, or a shortened version of, the local name assigned to the device. The Local Name data type value indicates if the name is complete or shortened. If the name is shortened, the complete name can be read using the remote name request procedure over BR/EDR or by reading the device name characteristic after the connection has been established using GATT.

A shortened name shall only contain contiguous characters from the beginning of the full name. For example, if the device name is ‘BT_Device_Name’ then the shortened name could be ‘BT_Device’ or ‘BT_Dev’.

1.2.2 Format

Data Type	Description
<<Shortened Local Name>>	Shortened local name
<<Complete Local Name>>	Complete local name

Table 1.3: Local Name Data Types

1.3 FLAGS

1.3.1 Description

The Flags data type contains one bit Boolean flags. The Flags data type shall be included when any of the Flag bits are non-zero and the advertising packet is connectable, otherwise the Flags data type may be omitted. All 0x00 octets after the last non-zero octet shall be omitted from the value transmitted.

Note: If the Flags AD type is not present in a non-connectable advertisement, the Flags should be considered as unknown and no assumptions should be made by the scanner.

Flags used over the LE physical channel are:

- Limited Discoverable Mode
- General Discoverable Mode
- BR/EDR Not Supported
- Simultaneous LE and BR/EDR to Same Device Capable (Controller)
- Simultaneous LE and BR/EDR to Same Device Capable (Host)

The LE Limited Discoverable Mode and LE General Discoverable Mode flags shall be ignored when received over the BR/EDR physical channel. The ‘BR/EDR Not Supported’ flag shall be set to 0 when sent over the BR/EDR physical channel.

1.3.2 Format

The Flags field may be zero or more octets long. This allows the Flags field to be extended while using the minimum number of octets within the data packet.

Data Type	Octet	Bit	Description
<<Flags>>	0	0	LE Limited Discoverable Mode
	0	1	LE General Discoverable Mode
	0	2	BR/EDR Not Supported. Bit 37 of LMP Feature Mask Definitions (Page 0)
	0	3	Simultaneous LE and BR/EDR to Same Device Capable (Controller). Bit 49 of LMP Feature Mask Definitions (Page 0)
	0	4	Simultaneous LE and BR/EDR to Same Device Capable (Host). Bit 66 of LMP Feature Mask Definitions (Page 1)
	0	5..7	Reserved for future use

Table 1.4: Flags Data Types

1.4 MANUFACTURER SPECIFIC DATA

1.4.1 Description

The Manufacturer Specific data type is used for manufacturer specific data. The first two data octets shall contain a company identifier code from the Assigned Numbers - Company Identifiers document. The interpretation of any other octets within the data shall be defined by the manufacturer specified by the company identifier.

1.4.2 Format

Data Type	Description
<<Manufacturer Specific Data>>	Size: 2 or more octets The first 2 octets contain the Company Identifier Code followed by additional manufacturer specific data

Table 1.5: Manufacturer Specific Data Type

1.5 TX POWER LEVEL

1.5.1 Description

The TX Power Level data type indicates the transmitted power level of the packet containing the data type. The TX Power Level should be the radiated power level. The TX Power Level data type may be used to calculate path loss on a received packet using the following equation:

pathloss = Tx Power Level – RSSI

where “RSSI” is the received signal strength, in dBm, of the packet received.

For example, if Tx Power Level = +4 (dBm) and the RSSI on the received packet is -60 (dBm) then the total path loss is +4 – (-60) = +64 dB. If a second packet were received at -40 dBm with a Tx Power Level data type = +15 dBm the resulting pathloss would be +55 dB. An application could use these pathloss values to choose which device it thinks might be closer (the one with the lower pathloss value).

Unfortunately, due to fading and varying antenna, circuit, and chip characteristics, these resulting pathloss values will have uncertainty. Some of the uncertainty (for example, due to fading) may be able to be removed if multiple packets are received from the same device.

Note: When the TX Power Level data type is not present, the TX power level of the packet is unknown.

1.5.2 Format

Data Type	Description
<<TX Power Level>>	Size: 1 octet 0xXX: -127 to +127 dBm

Table 1.6: TX Power Level Data Type

1.6 SECURE SIMPLE PAIRING OUT OF BAND (OOB)

1.6.1 Description

The Secure Simple Pairing Out of Band data types enable an out of band mechanism to communicate discovery information as well as other information related to the pairing process.

1.6.2 Format

The Secure Simple Pairing Out of Band data types shall be encapsulated in a OOB data block as defined in *Volume 3, Part C, section 5.2.2.7* and *Figure 5.6* of that section. The OOB data block consists of the mandatory part with fields SSP OOB Length and BD_ADDR as described in Table 1.7, followed by optional data types described in Table 1.8.

Field	Description
<<SSP OOB Length>>	Size: 2 octets 0xXXXX: 8 to 65535 bytes This field contains the length of the entire OOB data block including the length field itself.
<<BD_ADDR>>	Size: 6 octets Format defined in *[Vol. 2, Part B] Section 1.2*

Table 1.7: Fields for OOB Data Block Mandatory Part

Data Type	Description
<<Class of Device>>	Size: 3 octets Format defined in Assigned Numbers
<<Simple Pairing Hash C-192>>	Size: 16 octets Format defined in *[Vol. 2], Part H Section 7.2.2*
<<Simple Pairing Randomizer R-192>>	Size: 16 octets Format defined in *[Vol. 2], Part H Section 7.2.2*

Table 1.8: Data Types for OOB Data Block Optional Parts

Data Type	**Description**
<<Simple Pairing Hash C-256>>	Size: 16 octets Format defined in *[Vol. 2], Part H, Section 7.2.2*
<<LE Secure Connections Confirmation Value>>	Size: 16 octets Format defined in *[Vol 3], Part H, Section 2.3.5.6.4*
<<Simple Pairing Randomizer R-256>>	Size: 16 octets Format defined in *[Vol. 2], Part H, Section 7.2.2*
<<LE Secure Connections Random Value>>	Size: 16 octets Format defined in *[Vol 3], Part H, Section 2.3.5.6.4*

Table 1.8: Data Types for OOB Data Block Optional Parts

1.7 SECURITY MANAGER OUT OF BAND (OOB)

1.7.1 Description

The Security Manager Out of Band data type allows an out of band mechanism to be used by the Security Manager to communicate discovery information as well as other information related to the pairing process.

1.7.2 Format

The Security Manager Out of Band data type size is 1 octet.

Data Type	**Bit**	**Description**
<<Security Manager Out of Band Flag>>	0	OOB Flags Field (0 = OOB data not present, 1 = OOB data present)
	1	LE supported (Host) (i.e. bit 65 of LMP Extended Feature bits Page 1
	2	Simultaneous LE and BR/EDR to Same Device Capable (Host) (i.e. bit 66 of LMP Extended Feature bits Page 1)
	3	Address type (0 = Public Address, 1 = Random Address)
	4..7	Reserved for future use

Table 1.9: Security Manager OOB Flags Data Type

1.8 SECURITY MANAGER TK VALUE

1.8.1 Description

The Security Manager TK Value data type allows an out of band mechanism to be used by the Security Manager to communicate the TK value.

1.8.2 Format

Data Type	Description
<<Security Manager TK Value>>	Size: 16 octets Value as used in pairing over LE Physical channel. Format defined in *[Vol. 3], Part H Section 2.3*

Table 1.10: Security Manager TK Value Data Type

1.9 SLAVE CONNECTION INTERVAL RANGE

1.9.1 Description

The Slave Connection Interval Range data type contains the Peripheral's preferred connection interval range, for all logical connections. See *Vol 3, Part C, Section 12.3*.

Note: The minimum value depends on the battery considerations of the Peripheral and the maximum connection interval depends on the buffers available on the Peripheral.

The Central should use the information from the Peripheral's Slave Connection Interval Range data type when establishing a connection.

Note: Central and Peripheral are GAP roles as defined in *Vol.3, Part C, Section 2.2.2*.

1.9.2 Format

Data Type	Description
<<Slave Connection Interval Range>>	Size: 4 Octets The first 2 octets defines the minimum value for the connection interval in the following manner: $connInterval_{min}$ = Conn_Interval_Min * 1.25 ms Conn_Interval_Min range: 0x0006 to 0x0C80 Value of 0xFFFF indicates no specific minimum. Values not defined above are reserved for future use. The other 2 octets defines the maximum value for the connection interval in the following manner: $connInterval_{max}$ = Conn_Interval_Max * 1.25 ms Conn_Interval_Max range: 0x0006 to 0x0C80 Conn_Interval_Max shall be equal to or greater than the Conn_Interval_Min. Value of 0xFFFF indicates no specific maximum. Values not defined above are reserved for future use.

Table 1.11: Slave Connection Interval Range Data Type

1.10 SERVICE SOLICITATION

1.10.1 Description

A Peripheral device may send the Service Solicitation data type to invite Central devices that expose one or more of the services specified in the Service Solicitation data to connect. The Peripheral device should be in the undirected connectable mode and in one of the discoverable modes. This enables a Central device providing one or more of these services to connect to the Peripheral device, so that the Peripheral device can use the services on the Central device.

Note: Central and Peripheral are GAP roles as defined in *Vol.3, Part C, Section 2.2.2*.

1.10.2 Format

Data Type	Description
<<List of 16 bit Service Solicitation UUIDs>>	List of 16 bit Service Solicitation UUIDs
<<List of 32 bit Service Solicitation UUIDs>>	List of 32 bit Service Solicitation UUIDs
<<List of 128 bit Service Solicitation UUIDs>>	List of 128 bit Service Solicitation UUIDs

Table 1.12: Service Solicitation UUID Data Types

1.11 SERVICE DATA

1.11.1 Description

The Service Data data type consists of a service UUID with the data associated with that service.

1.11.2 Format

Data Type	**Description**
<<Service Data - 16 bit UUID>>	Size: 2 or more octets The first 2 octets contain the 16 bit Service UUID followed by additional service data
<<Service Data - 32 bit UUID>>	Size: 4 or more octets The first 4 octets contain the 32 bit Service UUID followed by additional service data
<<Service Data - 128 bit UUID>>	Size: 16 or more octets The first 16 octets contain the 128 bit Service UUID followed by additional service data

Table 1.13: Service Data

1.12 APPEARANCE

1.12.1 Description

The Appearance data type defines the external appearance of the device.

This value shall be the same as the Appearance characteristic, as defined in *Vol. 3, Part C, Section 12.2*.

1.12.2 Format

Data Type	**Description**
<<Appearance>>	The Appearance value shall be the enumerated value as defined by Bluetooth Assigned Numbers.

Table 1.14: Appearance

1.13 PUBLIC TARGET ADDRESS

1.13.1 Description

The Public Target Address data type defines the address of one or more intended recipients of an advertisement when one or more devices were bonded using a public address. This data type is intended to be used to avoid a situation where a bonded device unnecessarily responds to an advertisement intended for another bonded device.

1.13.2 Format

Data Type	Description
<<Public Target Address>>	Size: Multiples of 6 octets The format of each 6 octet address is the same as the Public Device Address defined in *Vol. 6, Part B, Section 1.3*. The Public Target Address value shall be the enumerated value as defined by Bluetooth Assigned Numbers.

Table 1.15: Public Target Address

1.14 RANDOM TARGET ADDRESS

1.14.1 Description

The Random Target Address data type defines the address of one or more intended recipients of an advertisement when one or more devices were bonded using a random address. This data type is intended to be used to avoid a situation where a bonded device unnecessarily responds to an advertisement intended for another bonded device.

1.14.2 Format

Data Type	Description
<<Random Target Address>>	Size: Multiples of 6 octets The format of each 6 octet address is the same as the Random Device Address defined in *Vol. 6, Part B, Section 1.3*. The Random Target Address value shall be the enumerated value as defined by Bluetooth Assigned Numbers.

Table 1.16: Random Target Address

1.15 ADVERTISING INTERVAL

1.15.1 Description

The Advertising Interval data type contains the *advInterval* value as defined in the Core specification, *Volume 6, Part B, Section 4.4.2.2*.

1.15.2 Format

Data Type	Description
<<Advertising Interval>>	Size: 2 octets (UINT16) Units: 0.625 ms advInterval value

Table 1.17: Advertising Interval.

1.16 LE BLUETOOTH DEVICE ADDRESS

1.16.1 Description

The LE Bluetooth Device Address data type defines the device address of the local device and the address type on the LE transport.

1.16.2 Format

Data Type	Description
<<LE Bluetooth Device Address>>	Size: 7 octets. The format of the 6 least significant Octets is the same as the Device Address defined in *[Vol. 6], Part B, Section 1.3*. The least significant bit of the most significant octet defines if the Device Address is a Public Address or a Random Address. LSB = 1 Then Random Device Address. LSB = 0 Then Public Device Address. Bits 1 to 7 in the most significant octet are reserved for future use.

Table 1.18: Bluetooth Device Address

1.17 LE ROLE

1.17.1 Description

The LE Role data type defines the LE role capabilities of the device.

1.17.2 Format

The LE Role data type size is 1 octet.

Data Type	Value	Description
<<LE Role>>	0x00	Only Peripheral Role supported
	0x01	Only Central Role supported
	0x02	Peripheral and Central Role supported, Peripheral Role preferred for connection establishment
	0x03	Peripheral and Central Role supported, Central Role preferred for connection establishment
	0x04 – 0xFF	Reserved for future use

Table 1.19: LE Role Data Type

1.18 UNIFORM RESOURCE IDENTIFIER (URI)

1.18.1 Description

The URI data type allows the representation of a URI, as defined in IETF STD 66. The URI data type is encoded using UTF-8. To help with compression, the first UTF-8 code point in the URI data type value represents a scheme name string, as defined below. All other UTF-8 code points in the URI data type shall be appended to the decompressed scheme name string and the result forms the URI.

The mapping of scheme name strings to UTF-8 code points is defined in the Bluetooth SIG assigned numbers page. Only permanent and provisional schemes, as defined by the IETF (see http://www.iana.org/assignments/uri-schemes.html, shall be assigned a scheme name and corresponding code point. Note that, except for the special case of U+0001, the decompressed scheme name string includes the ":" that separates the scheme from the remainder (the "hier-part") of the URI.

The code point of U+0001 shall be used when the scheme used is not defined as either a permanent or provisional scheme. This code point maps to the empty scheme name string.

Note: When U+0001 is used, the actual scheme and ":" must be included in the remaining UTF-8 code points.

1.18.2 Format

Data Type	Description
<<URI>>	Scheme name string and URI as a UTF-8 string

Table 1.20: URI Data Type

1.19 LE SUPPORTED FEATURES

1.19.1 Description

The LE Supported Features data type defines the LE features supported by the device. All 0x00 octets after the last non-zero octet shall be omitted from the value transmitted.

1.19.2 Format

The LE Supported Features data type size is zero or more octets long. This allows the LE Supported Features to be represented while using the minimum number of octets within the data packet.

Data Type	Description
<<LE Supported Features>>	The format is the same as the FeatureSet defined in *[Vol. 6] Part B Section 4.6*.

Table 1.21: LE Supported Features Data Type

1.20 CHANNEL MAP UPDATE INDICATION

1.20.1 Description

The channel map (*channelMap*) used for periodic advertisements may be updated at any time by the advertiser. The advertiser can update the channel map by sending the Channel Map Update Indication data type in the extended header of the packet containing the AUX_SYNC_IND PDU. The advertiser's Host may provide an initial channel map using the LE Set Host Channel Classification HCI Command; however the advertiser's Controller can update the channels that were marked as unknown by the Host in the channel map based on channel assessments without being requested to by the Host. The Channel Map Update Indication data type shall only be present in the extended header of the packet containing the AUX_SYNC_IND PDU.

The channel map used before the instant is known as $channelMap_{OLD}$. The channel map contained in the Channel Map Update Indication data type and used at the instant and after, is known as $channelMap_{NEW}$.

The Instant field shall be used to indicate the *paEventCount* value when $channelMap_{NEW}$ shall apply; this value is called the instant.

Upon first transmission of the data type the advertiser should allow a minimum of 6 AUX_SYNC_IND PDUs before the instant occurs.

When the value of *paEventCount* in the SyncInfo field is equal to the Instant field, the $channelMap_{NEW}$ shall be the current *channelMap*. The *lastUnmappedChannel* shall not be reset. If the *unmappedChannel* is an unused channel, then the $channelMap_{NEW}$ will be used when remapping. The only parameter that changes is the *channelMap*.

The advertiser shall not send a new Channel Map Update Indication data type before the instant.

1.20.2 Format

The Channel Map Update Indication data type size is 7 octets.

Data Type	Octets	Description
<<Channel Map Update Indication>>	0-4	ChM
	5-6	Instant

Table 1.22: Channel Map Update Indication

The ChM field shall contain the channel map indicating *Used* and *Unused* data channels. The format of this field is identical to the ChM field in the CONNECT_IND PDU (see Core Specification, *Volume 6, Part B, Section 2.3.3.1*).

The Instant field shall be set to indicate the number of advertising events as described in Section 1.20.1.

2 EXAMPLES

The following sections include examples of EIR and Advertising Data Types.

2.1 HOST EXAMPLES

2.1.1 Example Extended Inquiry Response

This is an example extended inquiry response for a phone with PANU and Hands-free Audio Gateway:

Value	Notes
0x06	Length of this Data
0x09	<<Complete Local Name>>
0x50	'P'
0x68	'h'
0x6F	'o'
0x6E	'n'
0x65	'e'
0x05	Length of this Data
0x03	<<Complete list of 16-bit Service UUIDs>>
0x15	PANU service class UUID
0x11	
0x1F	Hands-free Audio Gateway service class UUID
0x11	
0x01	Length of this data
0x05	<<Complete list of 32-bit Service UUIDs>>
0x01	Length of this data
0x07	<<Complete list of 128-bit Service UUIDs>>
0x00	End of Data (Not transmitted over the air)

Table 2.1: Example extended inquiry response

2.1.2 Example Advertising Data - Complete Local Name

This is an example of advertising data with AD types:

Value	Notes
0x02	Length of this Data
0x01	<<Flags>>
0x01	LE Limited Discoverable Flag set
0x0A	Length of this Data
0x09	<<Complete local name>>
0x50	‘P’
0x65	‘e’
0x64	‘d’
0x6F	‘o’
0x6D	‘m’
0x65	‘e’
0x74	‘t’
0x65	‘e’
0x72	‘r’

Table 2.2: Example advertising data with AD types

2.1.3 Example Advertising Data - URI

This example represents an advertisement of the URI “http://www.bluetooth.com”.

Value	Notes
0x15	Length of this data
0x24	<<URI>>
0x16	UTF-8 code point for “http:”
0x2F	‘/’
0x2F	‘/’
0x77	‘w’
0x77	‘w’
0x77	‘w’
0x2E	‘.’
0x62	‘b’
0x6C	‘l’
0x75	‘u’
0x65	‘e’
0x74	‘t’
0x6F	‘o’
0x6F	‘o’
0x74	‘t’
0x68	‘h’
0x2E	‘.’
0x63	‘c’
0x6F	‘o’
0x6D	‘m’

Table 2.3: Example advertising data with a URI data type for http://www.bluetooth.com

This example represents an advertisement of the URI "example://z.com/Ålborg".

Value	Notes
0x12	Length of this data
0x24	<<URI>>
0xC2	First UTF-8 octet for 'example:'
0xB9	Last UTF-8 octet for 'example:'
0x2F	'/'
0x2F	'/'
0x7A	'z'
0x2E	'.'
0x63	'c'
0x6F	'o'
0x6D	'm'
0x2F	'/'
0xC3	First UTF-8 octet for 'Å'
0x85	Last UTF-8 octet for 'Å'
0x6C	'l'
0x62	'b'
0x6F	'o'
0x72	'r'
0x67	'g'

Table 2.4: Example advertising data with a URI data type for example://z.com/Ålborg

2.2 CONTROLLER EXAMPLES

2.2.1 Example ACAD – Channel Map Update Indication

Value	Notes
0x08	Length of this Data
0x28	<<Channel Map Update Indication>>
0xFF	ChM = 0x1FFFFFF7FF
0xF7	
0xFF	
0xFF	
0x1F	
0x64	Instant = 0x0064
0x00	

Table 2.5: Example ACAD – Channel Map Update Indication

COMMON PROFILE AND SERVICE ERROR CODES

CONTENTS

1 OVERVIEW OF COMMON PROFILE AND SERVICE ERROR CODES

This document lists the common profile and service error codes sent over the Attribute Protocol. Error codes have a size of one octet.

1.1 USAGE DESCRIPTIONS

The purpose of this section is to give descriptions of how the common profile error codes should be used. It is beyond the scope of this document to give detailed descriptions of all situations where error codes can be used, especially as this is implementation dependent.

1.2 LIST OF ERROR CODES

The possible range of common profile error codes is 0xE0-0xFF. The Common Profile and Service Error Code Descriptions Part provides an error code usage description for each failure error code.

Values marked as “Reserved for Future Use”, can be used in future versions of the specification.

Error Code	Name
0xE0 – 0xFB	Reserved for Future Use
0xFC	Write Request Rejected
0xFD	Client Characteristic Configuration Descriptor Improperly Configured
0xFE	Procedure Already in Progress
0xFF	Out of Range

Table 1.1: List of Common Profile and Service Error Codes

2 COMMON PROFILE AND SERVICE ERROR CODE DESCRIPTIONS

2.1 OUT OF RANGE (0xFF)

The Out of Range error code is used when an attribute value is out of range as defined by a profile or service specification.

2.2 PROCEDURE ALREADY IN PROGRESS (0xFE)

The Procedure Already in Progress error code is used when a profile or service request cannot be serviced because an operation that has been previously triggered is still in progress.

2.3 CLIENT CHARACTERISTIC CONFIGURATION DESCRIPTOR IMPROPERLY CONFIGURED (0xFD)

The Client Characteristic Configuration Descriptor Improperly Configured error code is used when a Client Characteristic Configuration descriptor is not configured according to the requirements of the profile or service.

2.4 WRITE REQUEST REJECTED (0xFC)

The Write Request Rejected error code is used when a requested write operation cannot be fulfilled for reasons other than permissions.
Note: This differs from the “Write Not Permitted” error response in *Vol 3, Part F, Section 3.4.1.1* (ATT), which is intended when the write operation cannot be fulfilled due to permissions.

SERVICES PERMITTED TO USE SECURITY MODE 4 LEVEL 0

CONTENTS

1 SERVICES PERMITTED TO USE SECURITY MODE 4 LEVEL 0

The following sections enumerate the services permitted to use Security Mode 4 Level 0 as defined in *Volume 3, Part C, of the Bluetooth Core Specification,* v2.1 + EDR and later.

Section 1.1 enumerates those services that are permitted to use Security Mode 4 Level 0 over L2CAP connection oriented channels and Section 1.2 enumerates those services that are permitted to use Security Mode 4 Level 0 for unicast traffic over the L2CAP connectionless data channel (CID 0x0002).

Note: Security Mode 4 does not address broadcast traffic and hence this section is not relevant to broadcast data sent over the L2CAP connectionless data channel.

1.1 SECURITY MODE 4 LEVEL 0 OVER L2CAP CONNECTION-ORIENTED CHANNELS

Services corresponding to the following UUIDs may use Security Mode 4 Level 0 over an L2CAP connection-oriented channel when operated over a BR/EDR physical link. See *Bluetooth Core Specification Volume 3, Part B, Section 2.5.1* for more information on UUIDs. Also see Bluetooth Assigned Numbers for a list of assigned Service Class UUIDs.

- 0x1000 + Bluetooth_Base_UUID (Service Discovery Server)

1.2 SECURITY MODE 4 LEVEL 0 OVER THE L2CAP CONNECTIONLESS DATA CHANNEL

Services corresponding to the following UUIDs may use Security Mode 4 Level 0 for unicast traffic over the L2CAP connectionless data channel (CID 0x0002) when operated over a BR/EDR physical link. See *Bluetooth Core Specification Volume 3, Part B, Section 2.5.1* for more information on UUIDs. Also see Bluetooth Assigned Numbers for a list of assigned Service Class UUIDs.

- 3D_Display + Bluetooth_Base_UUID
  (3D Display service as defined in the 3D Synchronization Profile)
- 3D_Glasses + Bluetooth_Base_UUID
  (3D Glasses service as defined in the 3D Synchronization Profile)

unthinkably connected