Specification of the **Bluetooth®** System

Supplement to the Bluetooth Core Specification

CSS Version: 6

Adoption Date: 14 July 2015

Disclaimer and Copyright Notice

This disclaimer applies to all draft specifications and final specifications adopted by the Bluetooth SIG Board of Directors (both of which are hereinafter referred to herein as a Bluetooth “Specification”). Your use of this Specification in any way is subject to your compliance with all conditions of such use, and your acceptance of all disclaimers and limitations as to such use, contained in this Specification. Any user of this Specification is advised to seek appropriate legal, engineering or other professional advice regarding the use, interpretation or effect of this Specification on any matters discussed in this Specification.

Use of Bluetooth Specifications and any related intellectual property is governed by the Promoters Membership Agreement among the Promoter Members and Bluetooth SIG (the “Promoters Agreement”), certain membership agreements between Bluetooth SIG and its Adopter and Associate Members, including, but not limited to, the Membership Application, the Bluetooth Patent/Copyright License Agreement and the Bluetooth Trademark License Agreement (collectively, the “Membership Agreements”) and the Bluetooth Specification Early Adopters Agreements (1.2 Early Adopters Agreements) among Early Adopter members of the unincorporated Bluetooth SIG and the Promoter Members (the “Early Adopters Agreement”). Certain rights and obligations of the Promoter Members under the Early Adopters Agreements have been assigned to Bluetooth SIG by the Promoter Members.

Use of the Specification by anyone who is not a member of Bluetooth SIG or a party to an Early Adopters Agreement (each such person or party, a “Member”) is prohibited. The use of any portion of a Bluetooth Specification may involve the use of intellectual property rights ("IPR"), including pending or issued patents, or copyrights or other rights. Bluetooth SIG has made no search or investigation for such rights and disclaims any undertaking or duty to do so. The legal rights and obligations of each Member are governed by the applicable Membership Agreements, Early Adopters Agreement or Promoters Agreement. No license, express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein.

Any use of the Specification not in compliance with the terms of the applicable Membership Agreements, Early Adopters Agreement or Promoters Agreement is prohibited and any such prohibited use may result in (i) termination of the applicable Membership Agreements or Early Adopters Agreement and (ii) liability claims by Bluetooth SIG or any of its Members for patent, copyright and/or trademark infringement claims permitted by the applicable agreement or by applicable law.

THE SPECIFICATION IS PROVIDED “AS IS” WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, SATISFACTORY QUALITY, OR REASONABLE SKILL OR CARE, OR ANY WARRANTY ARISING OUT OF ANY COURSE OF DEALING, USAGE, TRADE PRACTICE, PROPOSAL, SPECIFICATION OR SAMPLE.

Each Member hereby acknowledges that products equipped with the Bluetooth wireless technology ("Bluetooth Products") may be subject to various regulatory controls under the laws and regulations applicable to products using wireless non licensed spectrum of various governments worldwide. Such laws and regulatory controls may govern, among other things, the combination, operation, use, implementation and distribution of Bluetooth Products. Examples of such laws and regulatory controls include, but are not limited to, airline regulatory controls, telecommunications regulations, technology transfer controls and health and safety regulations. Each Member is solely responsible for the compliance by their Bluetooth Products with any such laws and regulations and for obtaining any and all required authorizations, permits, or licenses for their Bluetooth Products related to such regulations within the applicable jurisdictions. Each Member acknowledges that nothing in the Specification provides any information or assistance in connection with securing such compliance, authorizations or licenses. NOTHING IN THE SPECIFICATION CREATES ANY WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING SUCH LAWS OR REGULATIONS.

ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHTS OR FOR NONCOMPLIANCE WITH LAWS, RELATING TO USE OF THE SPECIFICATION IS EXPRESSLY DISCLAIMED. To the extent not prohibited by law, in no event will Bluetooth SIG or its Members or their affiliates be liable for any damages, including without limitation, lost revenue, profits, data or programs, or business interruption, or for special, indirect, consequential, incidental or punitive damages, however caused and regardless of the theory of liability, arising out of or related to any furnishing, practicing, modifying, use or the performance or implementation of the contents of this Specification, even if Bluetooth SIG or its Members or their affiliates have been advised of the possibility of such damages. BY USE OF THE SPECIFICATION, EACH MEMBER EXPRESSLY WAIVES ANY CLAIM AGAINST BLUETOOTH SIG AND ITS MEMBERS OR THEIR AFFILATES RELATED TO USE OF THE SPECIFICATION.

If this Specification is an intermediate draft, it is for comment only. No products should be designed based on it except solely to verify the prototyping specification at SIG sponsored IOP events and it does not represent any commitment to release or implement any portion of the intermediate draft, which may be withdrawn, modified, or replaced at any time in the adopted Specification.

Bluetooth SIG reserves the right to adopt any changes or alterations to the Specification it deems necessary or appropriate.

TABLE OF CONTENTS

Part A
DATA TYPES SPECIFICATION

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), and OOB data blocks. All data types defined here may be used for EIR or AD data types unless stated otherwise. Additional data types may be defined in profile specifications.

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.

An extended inquiry response or advertising data packet 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 1.1: 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.

An extended inquiry response packet or advertising data packet shall not contain more than one instance of the Local Name data type.

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 1.2: 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 octets that are 0x00 are not transmitted as long as all other octets after that octet are also 0x00.

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.

An extended inquiry response packet or advertising data packet shall not contain more than one instance of the Flags data type.

The Flags AD type shall not be included in the scan response data.

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

Table 1.3: 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 1.4: 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 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 1.5: 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.

The Secure Simple Pairing Out of Band data types shall not be used in EIR or AD packets over the BR/EDR or LE transports and shall be only used over an out-of-band mechanism.

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 1.6, followed by optional data types described in Table 1.7.

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 1.6: 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*
<<Simple Pairing Hash C-256>>	Size: 16 octets Format defined in *[Vol. 2], Part H, Section 7.2.2*

Table 1.7: Data Types for OOB Data Block Optional Parts

Data Type	Description
<<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 1.7: 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.

The Security Manager Out of Band data type shall not be used in EIR or AD packets over the BR/EDR or LE transports and may only be sent over an out-of-band mechanism.

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

Table 1.8: 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.

The Security Manager TK Value data type shall not be used in EIR or AD packets over the BR/EDR or LE transports and shall only be sent over an out-of-band mechanism.

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 1.9: 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 12.5*.

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.

This data type shall not be sent over EIR.

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. 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.

Table 1.10: 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.

The Service Solicitation data type shall not be sent over EIR.

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 1.11: 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.

The Service Data data type shall not be sent over EIR.

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 1.12: Service Data

1.12 APPEARANCE

1.12.1 Description

The Appearance data type defines the external appearance of the device. The Appearance data type shall exist only once. It may be sent in either the Advertising or Scan Response data, but not both.

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

The Appearance data type shall not be sent over EIR.

1.12.2 Format

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

Table 1.13: 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.

This data type shall exist only once. It may be sent in either the Advertising or Scan Response data, but not both.

This data type shall not be sent over EIR.

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 1.14: 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.

This data type shall exist only once. It may be sent in either the Advertising or Scan Response data, but not both.

This data type shall not be sent over EIR.

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 1.15: 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*.

An Advertising data field shall not contain more than one instance of the Advertising Interval data type.

1.15.2 Format

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

Table 1.16: 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. This data type shall exist only once. It may be sent by an out of band method to a peer device.

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 1.17: Bluetooth Device Address

1.17 LE ROLE

1.17.1 Description

The LE Role data type defines the LE role capabilities of the device. This data type shall exist only once. It may be sent by an out of band method to a peer 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 1.18: 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 1.19: URI Data Type

2 EXAMPLES

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

2.1 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

2.2 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

2.3 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

Supplement to Bluetooth Core Specification
Part B

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 – 0xFC	Reserved for Future Use
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.

Supplement to Bluetooth Core Specification
Part C

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)