



Supplement to the Bluetooth Core Specification

Bluetooth® Specification

- **Version:** v13
- **Version Date:** 2025-04-29
- **Group Prepared By:** Core Specification Working Group [E26111_r02]

Abstract:

This supplement contains information related to data types, common Profile and Service error codes, and enumerates the services permitted to use Security Mode 4 Level 0.





Disclaimer and Copyright Notice

Use of this specification is your acknowledgement that you agree to and will comply with the following notices and disclaimers. You are advised to seek appropriate legal, engineering, and other professional advice regarding the use, interpretation, and effect of this specification.

Use of Bluetooth specifications by members of Bluetooth SIG is governed by the membership and other related agreements between Bluetooth SIG and its members, including those agreements posted on Bluetooth SIG's website located at www.bluetooth.com. Any use of this specification by a member that is not in compliance with the applicable membership and other related agreements is prohibited and, among other things, may result in (i) termination of the applicable agreements and (ii) liability for infringement of the intellectual property rights of Bluetooth SIG and its members. This specification may provide options, because, for example, some products do not implement every portion of the specification. All content within the specification, including notes, appendices, figures, tables, message sequence charts, examples, sample data, and each option identified is intended to be within the bounds of the Scope as defined in the Bluetooth Patent/Copyright License Agreement ("PCLA"). Also, the identification of options for implementing a portion of the specification is intended to provide design flexibility without establishing, for purposes of the PCLA, that any of these options is a "technically reasonable non-infringing alternative."

Use of this specification by anyone who is not a member of Bluetooth SIG is prohibited and is an infringement of the intellectual property rights of Bluetooth SIG and its members. The furnishing of this specification does not grant any license to any intellectual property of Bluetooth SIG or its members. THIS SPECIFICATION IS PROVIDED "AS IS" AND BLUETOOTH SIG, ITS MEMBERS AND THEIR AFFILIATES MAKE NO REPRESENTATIONS OR WARRANTIES AND DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, TITLE, NON-INFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR THAT THE CONTENT OF THIS SPECIFICATION IS FREE OF ERRORS. For the avoidance of doubt, Bluetooth SIG has not made any search or investigation as to third parties that may claim rights in or to any specifications or any intellectual property that may be required to implement any specifications and it disclaims any obligation or duty to do so.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, BLUETOOTH SIG, ITS MEMBERS AND THEIR AFFILIATES DISCLAIM ALL LIABILITY ARISING OUT OF OR RELATING TO USE OF THIS SPECIFICATION AND ANY INFORMATION CONTAINED IN THIS SPECIFICATION, INCLUDING 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, AND EVEN IF BLUETOOTH SIG, ITS MEMBERS OR THEIR AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF THE DAMAGES.

Products equipped with Bluetooth wireless technology ("Bluetooth Products") and their combination, operation, use, implementation, and distribution may be subject to regulatory controls under the laws and regulations of numerous countries that regulate products that use wireless non-licensed spectrum. Examples include airline regulations, telecommunications regulations, technology transfer controls, and health and safety regulations. You are solely responsible for complying with all applicable laws and regulations and for obtaining any and all required authorizations, permits, or licenses in connection with your use of this specification and development, manufacture, and distribution of Bluetooth Products. Nothing in this specification provides any information or assistance in connection with complying with applicable laws or regulations or obtaining required authorizations, permits, or licenses.

Bluetooth SIG is not required to adopt any specification or portion thereof. If this specification is not the final version adopted by Bluetooth SIG's Board of Directors, it may not be adopted. Any specification adopted by Bluetooth SIG's Board of Directors may be withdrawn, replaced, or modified at any time. Bluetooth SIG reserves the right to change or alter final specifications in accordance with its membership and operating agreements.

Copyright © 1999–2024⁵. All copyrights in the Bluetooth Specifications themselves are owned by Apple Inc., Ericsson AB, Intel Corporation, [Google LLC](#), Lenovo (Singapore) Pte. Ltd., Microsoft Corporation, Nokia Corporation, and Toshiba Corporation. The Bluetooth word mark



and logos are owned by Bluetooth SIG, Inc. Other third-party brands and names are the property of their respective owners.

Language and Other Conventions

The conventions in [Vol 1] Part E apply to the Supplement to the Bluetooth Core Specification.

Version History

Revision	Date	Comment
v1	2011-12-27	Initial version.
v2	2012-07-24	Add Common Profile and Service Error Codes.
v3	2013-02-12	Add Advertising Interval data type. Add Services Permitted to Use Security Mode 4 Level 0.
v4	2013-12-03	Add LE Bluetooth Device Address data type. Add LE Role data type. Add BR/EDR Secure Connections support. Adopted along with Core v4.1.
v5	2014-12-02	Add LE Secure Connections support. Adopted along with Core v4.2.
v6	2015-07-14	Add Uniform Resource Identifier (URI) data type.
v7	2016-12-06	Add Write Request Rejected Common Profile and Service Error Code. Add LE Supported Features data type. Add Channel Map Update Indication data type. Add Table 1.1 that contains requirements on where the different data types can appear. Errata integrated: 6528, 6610, 6695, 7008, 7019, 7117, 7377, 7510, 8057 Adopted along with Core v5.0.
v8	2019-01-21	Errata integrated: 8634, 8635 Issues integrated: 9540, 9988, 10280, 10910, 11020 Adopted along with Core v5.1.
v9	2019-12-31	Add BIGInfo data type. Add Broadcast_Code data type. Erratum integrated: 11533 Issue integrated: 11019 Adopted along with Core v5.2.



Revision	Date	Comment
v10	2021-07-13	Errata integrated: 12064, 13416, 15039, 15064, 15498, 15583 Issues integrated: 12596, 13592, 15851, 16671 Updated terminology to remove inappropriate language. ¹ Adopted along with Core v5.3.
v11	2023-01-31	Errata integrated: 17332, 17430, 17888, 17926, 18737, 20357 Issues integrated: 18566, 18567, 18835, 18989, 19026, 19043, 19070, 22281, 22374, 22445 Add Encrypted Data data type. Add Periodic Advertising with Responses. Adopted along with Core v5.4.
v12	2024-08-27	Errata integrated: 22527, 22531, 23635, 24039 Issues integrated: 22525, 22529, 22530, 25021 Adopted along with Core v6.0.
<u>v13</u>	<u>2025-05-05</u>	<u>Errata integrated: 25277, 26111</u> <u>Adopted along with Core v6.1.</u>

1. Certain terms that were identified as inappropriate have been replaced. As a consequence, some other terms have also been changed to retain consistency and some HCI command and event names had consequential changes. For a list of the original terms and names and their replacements, see the Appropriate Language Mapping Tables, <https://www.bluetooth.com/language-mapping/Appropriate-Language-Mapping-Table>.



TABLE OF CONTENTS

Part A

DATA TYPES SPECIFICATION

1	Data Types definitions and formats	12
1.1	Service or Service Class UUID	13
1.1.1	Description	13
1.1.2	Format	14
1.2	Local name	14
1.2.1	Description	14
1.2.2	Format	14
1.3	Flags	15
1.3.1	Description	15
1.3.2	Format	15
1.4	Manufacturer Specific Data	16
1.4.1	Description	16
1.4.2	Format	16
1.5	TX Power Level	16
1.5.1	Description	16
1.5.2	Format	17
1.6	Secure Simple Pairing Out of Band (OOB)	17
1.6.1	Description	17
1.6.2	Format	17
1.7	Security Manager Out of Band (OOB)	18
1.7.1	Description	18
1.7.2	Format	18
1.8	Security Manager TK Value	18
1.8.1	Description	18
1.8.2	Format	18
1.9	Peripheral Connection Interval Range	18
1.9.1	Description	18
1.9.2	Format	19
1.10	Service Solicitation	19
1.10.1	Description	19
1.10.2	Format	20
1.11	Service Data	20
1.11.1	Description	20
1.11.2	Format	20
1.12	Appearance	20



1.12.1	Description	20
1.12.2	Format.....	21
1.13	Public Target Address	21
1.13.1	Description	21
1.13.2	Format.....	21
1.14	Random Target Address	21
1.14.1	Description	21
1.14.2	Format	21
1.15	Advertising Interval	22
1.15.1	Description	22
1.15.2	Format	22
1.16	LE Bluetooth Device Address	22
1.16.1	Description	22
1.16.2	Format	22
1.17	LE Role.....	22
1.17.1	Description	22
1.17.2	Format.....	23
1.18	Uniform Resource Identifier (URI)	23
1.18.1	Description	23
1.18.2	Format	23
1.19	LE Supported Features	24
1.19.1	Description	24
1.19.2	Format.....	24
1.20	Channel Map Update Indication	24
1.20.1	Description	24
1.20.2	Format.....	25
1.21	BIGInfo	25
1.21.1	Description	25
1.21.2	Format.....	25
1.22	Broadcast_Code.....	25
1.22.1	Description	25
1.22.2	Format.....	25
1.23	Encrypted Data.....	26
1.23.1	Description	26
1.23.2	Format.....	26
1.23.3	Encryption algorithm	26
1.23.4	Security properties.....	26
1.24	Periodic Advertising Response Timing Information	27
1.24.1	Description	27



1.24.2	Format.....	28
2	Examples	29
2.1	Host Examples	29
2.1.1	Example extended inquiry response.....	29
2.1.2	Example advertising data – Complete Local Name	29
2.1.3	Example advertising data – URI	30
2.2	Controller Examples	32
2.2.1	Example ACAD – Channel Map Update Indication	32
2.3	Encrypted Advertising Data Sample Data	32
2.3.1	Encrypted Advertising Data Set 1	32
2.3.2	Encrypted Advertising Data Set 2	33

Part B

COMMON PROFILE AND SERVICE ERROR CODES

1	Overview of Common Profile and Service Error Codes	37
1.1	Usage Descriptions	37
1.2	List of Error Codes.....	37
2	Common Profile and Service Error Code Descriptions	38
2.1	Out of Range (0xFF).....	38
2.2	Procedure Already in Progress (0xFE)	38
2.3	Client Characteristic Configuration Descriptor Improperly Configured (0xFD)	38
2.4	Write Request Rejected (0xFC).....	38

Part C

SERVICES THAT MAY USE SECURITY MODE 4 LEVEL 0

1	Services That May Use Security Mode 4 Level 0	41
1.1	Security Mode 4 Level 0 over L2CAP Connection-oriented Channels	41
1.2	Security Mode 4 Level 0 over the L2CAP Connectionless Data Channel	41

Supplement to the Bluetooth Core Specification
Part A

**DATA TYPES
SPECIFICATION**



CONTENTS

1	Data Types definitions and formats	12
1.1	Service or Service Class UUID	13
1.1.1	Description	13
1.1.2	Format	14
1.2	Local name	14
1.2.1	Description	14
1.2.2	Format	14
1.3	Flags	15
1.3.1	Description	15
1.3.2	Format	15
1.4	Manufacturer Specific Data	16
1.4.1	Description	16
1.4.2	Format	16
1.5	TX Power Level	16
1.5.1	Description	16
1.5.2	Format	17
1.6	Secure Simple Pairing Out of Band (OOB)	17
1.6.1	Description	17
1.6.2	Format	17
1.7	Security Manager Out of Band (OOB)	18
1.7.1	Description	18
1.7.2	Format	18
1.8	Security Manager TK Value	18
1.8.1	Description	18
1.8.2	Format	18
1.9	Peripheral Connection Interval Range	18
1.9.1	Description	18
1.9.2	Format	19
1.10	Service Solicitation	19
1.10.1	Description	19
1.10.2	Format	20
1.11	Service Data	20
1.11.1	Description	20
1.11.2	Format	20
1.12	Appearance	20
1.12.1	Description	20
1.12.2	Format	21



1.13	Public Target Address	21
1.13.1	Description	21
1.13.2	Format	21
1.14	Random Target Address	21
1.14.1	Description	21
1.14.2	Format	21
1.15	Advertising Interval	22
1.15.1	Description	22
1.15.2	Format	22
1.16	LE Bluetooth Device Address	22
1.16.1	Description	22
1.16.2	Format	22
1.17	LE Role	22
1.17.1	Description	22
1.17.2	Format	23
1.18	Uniform Resource Identifier (URI)	23
1.18.1	Description	23
1.18.2	Format	23
1.19	LE Supported Features	24
1.19.1	Description	24
1.19.2	Format	24
1.20	Channel Map Update Indication	24
1.20.1	Description	24
1.20.2	Format	25
1.21	BIGInfo	25
1.21.1	Description	25
1.21.2	Format	25
1.22	Broadcast_Code	25
1.22.1	Description	25
1.22.2	Format	25
1.23	Encrypted Data	26
1.23.1	Description	26
1.23.2	Format	26
1.23.3	Encryption algorithm	26
1.23.4	Security properties	26
1.24	Periodic Advertising Response Timing Information	27
1.24.1	Description	27
1.24.2	Format	28
2	Examples	29

Data Types Specification

2.1	Host Examples	29
2.1.1	Example extended inquiry response	29
2.1.2	Example advertising data – Complete Local Name	29
2.1.3	Example advertising data – URI	30
2.2	Controller Examples	32
2.2.1	Example ACAD – Channel Map Update Indication	32
2.3	Encrypted Advertising Data Sample Data	32
2.3.1	Encrypted Advertising Data Set 1	32
2.3.2	Encrypted Advertising Data Set 2	33



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 <i>or</i> Service Class UUID [E25277_r01]	O	O	O	O	O
Local Name	C.1	C.1	C.1	X	C.1
Flags	C.1	C.1	X	X	C.1
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
Peripheral Connection Interval Range	X	O	O	X	O
Service Solicitation	X	O	O	X	O
Service Data	X	O	O	O	O
Appearance	X	C.2	C.2	X	C.1
Public Target Address	X	C.2	C.2	X	C.1
Random Target Address	X	C.2	C.2	X	C.1
Advertising Interval	X	C.1	C.1	X	C.1
LE Bluetooth Device Address	X	X	X	X	C.1
LE Role	X	X	X	X	C.1
Uniform Resource Identifier	O	O	O	X	O
LE Supported Features	X	C.1	C.1	X	C.1
Channel Map Update Indication	X	X	X	C.1	X
BIGInfo	X	X	X	C.1	X
Broadcast_Code	X	X	X	X	O

Table 1.1: Permitted usages for data types



Data type	Context				
	EIR	AD	SRD	ACAD	OOB
Encrypted Data	O	O	O	X	O
Periodic Advertising Response Timing Information	X	X	X	C.1	X

O: Optional in this context (may appear more than once in a block).
C.1: Optional in this context; shall not appear more than once in a block.
C.2: 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.

Table 1.1: Permitted usages for data types

The values for the data types are listed in [Assigned Numbers](#). In the format descriptions, the name of the data type (e.g., «Flags») is followed by the type of the value as defined in [Vol 1] Part E, Section 2.9.

1.1 SERVICE OR SERVICE CLASS UUID

[E25277_r01]

GAP and GATT service UUIDs should not be included in a Service or Service Class UUIDs AD type, for either a complete or incomplete list. [E25277_r01]

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 or Service Class UUIDs that may be returned: [E25277_r01]

- 16-bit Bluetooth Service or Service Class UUIDs [E25277_r01]
- 32-bit Bluetooth Service or Service Class UUIDs [E25277_r01]
- Global 128-bit Service or Service Class UUIDs [E25277_r01]

Two Service or Service Class UUID data types are assigned to each size of Service UUID. One Service or Service Class UUID data type indicates that the Service or Service Class UUID list is incomplete and the other indicates the Service or Service Class UUID list is complete. [E25277_r01]

A packet or data block shall not contain more than one instance for each Service or Service Class UUID data size. If a device has no Service or Service Class 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 or Service Class UUIDs. An omitted Service or



Service Class UUID data type shall be interpreted as an empty incomplete-list. [E25277_r01]

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 <u>or Service Class</u> UUIDs» <i>UUID16[]</i> [E25277_r01]	More 16-bit Service <u>or Service Class</u> UUIDs available [E25277_r01]
«Complete List of 16-bit Service <u>or Service Class</u> UUIDs» <i>UUID16[]</i> [E25277_r01]	Complete list of 16-bit Service <u>or Service Class</u> UUIDs [E25277_r01]
«Incomplete List of 32-bit Service <u>or Service Class</u> UUIDs» <i>UUID32[]</i> [E25277_r01]	More 32-bit Service <u>or Service Class</u> UUIDs available [E25277_r01]
«Complete List of 32-bit Service <u>or Service Class</u> UUIDs» <i>UUID32[]</i> [E25277_r01]	Complete list of 32-bit Service <u>or Service Class</u> UUIDs [E25277_r01]
«Incomplete List of 128-bit Service <u>or Service Class</u> UUIDs» <i>UUID128[]</i> [E25277_r01]	More 128-bit Service <u>or Service Class</u> UUIDs available [E25277_r01]
«Complete List of 128-bit Service <u>or Service Class</u> UUIDs» <i>UUID128[]</i> [E25277_r01]	Complete list of 128-bit Service <u>or Service Class</u> UUIDs [E25277_r01]

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» <i>utf8s</i>	Shortened local name
«Complete Local Name» <i>utf8s</i>	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 all-zero 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)

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	Bit	Description
«Flags» <i>boolean[]</i>	0	LE Limited Discoverable Mode
	1	LE General Discoverable Mode
	2	BR/EDR Not Supported. Bit 37 of LMP Feature Mask Definitions (Page 0)
	3	Simultaneous LE and BR/EDR to Same Device Capable (Controller). Bit 49 of LMP Feature Mask Definitions (Page 0)
	4	Previously Used

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 from [Assigned Numbers](#). 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» <i>uint16</i> , which may be followed by <i>struct</i>	The first value contains the Company Identifier Code. Any remainder contains 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. If the power level is included in a TxPower field (see [Vol 6] Part B, Section 2.3.4.7), then the Controller should set the value to be as accurate as possible. If the Controller is aware that the power level varies across frequencies, then it should update the value depending on the frequency that the packet is being sent on. If the power level is included in a «TX Power Level» AD Structure (see [Vol 3] Part C, Section 11) created by the Host, then the Host should set the value to be as accurate as possible.

The TX Power Level data type may be used to calculate path loss on a received packet using the following equation:

$$\text{pathloss} = \text{Tx Power Level} - \text{RSSI}$$

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» <i>sint8</i>	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 [Vol 3] Part C, Section 5.2.2.7. The optional data types in the OOB block are described in [Table 1.7](#).

Data Type	Description
«Class of Device» <i>uint24</i>	Format defined in Assigned Numbers
«Secure Simple Pairing Hash C-192» <i>uint128</i>	Commitment value specified in [Vol 2] Part H, Section 7.2.2.
«Secure Simple Pairing Randomizer R-192» <i>uint128</i>	Random number used in [Vol 2] Part H, Section 7.2.2.
«Secure Simple Pairing Hash C-256» <i>uint128</i>	Commitment value specified in [Vol 2] Part H, Section 7.2.2.
«Secure Simple Pairing Randomizer R-256» <i>uint128</i>	Random number used in [Vol 2] Part H, Section 7.2.2.
«LE Secure Connections Confirmation Value» <i>uint128</i>	Confirm value specified in [Vol 3] Part H, Section 2.3.5.6.4.
«LE Secure Connections Random Value» <i>uint128</i>	Random number used 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.

1.7.2 Format

Data Type	Bit	Description
«Security Manager Out of Band Flag» <i>boolean[8]</i>	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	Previously Used
	3	Address type (0 = Public Address, 1 = Random Address)

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.

1.8.2 Format

Data Type	Description
«Security Manager TK Value» <i>uint128</i>	Temporary key used in pairing over LE Physical channel; see [Vol 3] Part H, Section 2.3.

Table 1.9: Security Manager TK Value data type

1.9 PERIPHERAL CONNECTION INTERVAL RANGE

1.9.1 Description

The Peripheral 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 Connection Interval Range data provided by the Peripheral 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
«Peripheral Connection Interval Range» <i>uint16[2]</i>	<p>The first value defines the minimum value for the connection interval in the following manner:</p> $\text{connInterval}_{min} = \text{Conn_Interval_Min} \times 1.25 \text{ ms}$ <p>Conn_Interval_Min range: 0x0006 to 0x0C80 Value of 0xFFFF indicates no specific minimum. Values not defined above are reserved for future use.</p> <p>The second value defines the maximum value for the connection interval in the following manner:</p> $\text{connInterval}_{max} = \text{Conn_Interval_Max} \times 1.25 \text{ ms}$ <p>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.</p>

Table 1.10: Peripheral Connection Interval Range data type

1.10 SERVICE SOLICITATION

1.10.1 Description

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

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» <i>UUID16[]</i>	List of 16 bit Service Solicitation UUIDs
«List of 32 bit Service Solicitation UUIDs» <i>UUID32[]</i>	List of 32 bit Service Solicitation UUIDs
«List of 128 bit Service Solicitation UUIDs» <i>UUID128[]</i>	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.

1.11.2 Format

Data Type	Description
«Service Data - 16 bit UUID» <i>UUID16</i> , which may be followed by <i>struct</i>	The first value contains the 16 bit Service UUID. Any remainder contains additional service data.
«Service Data - 32 bit UUID» <i>UUID32</i> , which may be followed by <i>struct</i>	The first value contains the 32 bit Service UUID. Any remainder contains additional service data.
«Service Data - 128 bit UUID» <i>UUID128</i> , which may be followed by <i>struct</i>	The first value contains the 128 bit Service UUID. Any remainder contains additional service data.

Table 1.12: Service Data data types

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» <i>uint16</i>	The Appearance value shall be the enumerated value as defined by Assigned Numbers .

Table 1.13: Appearance data type

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» <i>uint48[]</i>	The format of each address is the same as the Public Device Address defined in [Vol 6] Part B, Section 1.3.

Table 1.14: Public Target Address data type

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» <i>uint48[]</i>	The format of each address is the same as the Random Device Address defined in [Vol 6] Part B, Section 1.3.

Table 1.15: Random Target Address data type



1.15 ADVERTISING INTERVAL

1.15.1 Description

The Advertising Interval data type contains the *advInterval* value as defined in [Vol 6] Part B, Section 4.4.2.2.

If *advInterval* is less than 40.96 s, the «Advertising Interval - long» data type shall not be used. If *advInterval* is 40.96 s or greater, the «Advertising Interval - long» data type shall be used.

1.15.2 Format

Data Type	Description
«Advertising Interval» <i>uint16</i>	Units: 0.625 ms <i>advInterval</i> value
«Advertising Interval - long» <i>uint24</i> or <i>uint32</i>	Units: 0.625 ms <i>advInterval</i> value

Table 1.16: Advertising Interval data type

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» <i>uint48</i> followed by <i>boolean</i>	The first value is the Device Address defined in [Vol 6] Part B, Section 1.3. The second value indicates whether the Device Address is a Public Address or a Random Address: 0 = Public Device Address 1 = Random Device Address.

Table 1.17: Bluetooth Device Address data type

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

Data Type	Value	Description
«LE Role» <i>uint8</i>	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 to 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 [Assigned Numbers](#). 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.

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.

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

1.18.2 Format

Data Type	Description
«URI» <i>utf8s</i>	Scheme name string and URI as a UTF-8 string

Table 1.19: 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 all-zero octets after the last non-zero octet shall be omitted from the value transmitted. This allows the LE Supported Features to be represented while using the minimum number of octets within the data packet.

1.19.2 Format

Data Type	Description
«LE Supported Features» <i>boolean[]</i>	Each value corresponds to the bit with the same position in the FeatureSet defined in [Vol 6] Part B, Section 4.6.

Table 1.20: 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 or AUX_SYNC_SUBEVENT_IND PDU. The advertiser's Host may provide an initial channel map using the HCI_LE_Set_Host_Channel_Classification 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 or AUX_SYNC_SUBEVENT_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 periodic advertising intervals before the instant occurs.

When the value of *paEventCount* is equal to the Instant field, the *channelMap_{NEW}* shall become 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

Data Type	Octets	Description
«Channel Map Update Indication» <i>boolean</i> [37] followed by <i>uint16</i>	0-4	ChM
	5-6	Instant

Table 1.21: Channel Map Update Indication data type

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 [Vol 6] Part B, Section 2.3.3.1).

The Instant field shall be set to indicate the instant as described in [Section 1.20.1](#).

1.21 BIGINFO

1.21.1 Description

The BIGInfo data type contains the necessary information for a Synchronized Receiver to synchronize to a BIG that is being broadcast by an Isochronous Broadcaster.

1.21.2 Format

The format for BIGInfo is described in [Vol 6] Part B, Section 4.4.6.11.

1.22 BROADCAST_CODE

1.22.1 Description

The Broadcast_Code data type contains the string format of the Broadcast_Code for an encrypted BIG.

1.22.2 Format

The format for Broadcast_Code is described in [Vol 3] Part C, Section 3.2.6. It should not include trailing all-zero octets.



1.23 ENCRYPTED DATA

1.23.1 Description

The Encrypted Data data type consists of an encrypted payload secured with a pre-shared session key, a pre-shared initialization vector, and randomizer. It is authenticated using a message integrity check (MIC). The session key and initialization vector can be shared as described in [Vol 3] Part C, Section 12.6.

1.23.2 Format

The Encrypted Data data type shall contain Randomizer, Payload, and MIC fields.

Data Type	Octets ¹	Description
«Encrypted Data» <i>uint20</i> followed by <i>uint8[]</i> followed by <i>uint8[4]</i>	0 to 4	Randomizer
	5 to L+4	Payload
	L+5 to L+8	MIC

Table 1.22: Encrypted Data data type

1. L is the length of the payload.

The Payload field shall contain a sequence of one or more AD structures (see [Vol 3] Part C, Section 11) that are encrypted as described in [Section 1.23.3](#).

1.23.3 Encryption algorithm

The Payload and MIC fields shall be encrypted using the CCM algorithm defined in [Vol 6] Part E, Section 1, with the following changes:

- The *packetCounter* of the CCM nonce shall be set to the Randomizer field of the AD Data. The *directionBit* of the CCM nonce shall be set to the most significant bit of the Randomizer field.
- The session key shall be set to a value determined by a higher layer specification or otherwise negotiated between the devices that are sending and receiving this AD type. Any session keys with at least 128 bits of entropy may be used.
- The initialization vector (IV) of the CCM nonce shall be set to values determined by a higher-level specification or otherwise negotiated between the devices that are sending and receiving this AD type.
- In the B₁ block, octet 2 shall equal 0xEA.

1.23.4 Security properties

The Randomizer field shall be changed each time the data in the Payload field is changed or the random device address is changed (see [Vol 3] Part C, Sec-



tion 10.7.1.1). The IV shall be changed each time the session key is changed. If the randomizer value is not generated using the requirements for random numbers defined in [Vol 2] Part H, Section 2, then the Host's privacy can be compromised; e.g., making it easier for an attacker to track a device over a period of time.

1.24 PERIODIC ADVERTISING RESPONSE TIMING INFORMATION

1.24.1 Description

The Periodic Advertising Response Timing Information data type contains additional synchronization information used for Periodic advertising with responses (PAwR) trains.



1.24.2 Format

Data Type	Description	Octets	Fields
«Periodic Advertising Response Timing Information» <i>uint32</i> followed by <i>uint8[4]</i>	Periodic advertising response timing information for the advertiser.	0 to 3	RspAA: Access Address to be used by the device when it transmits a response packet to the periodic advertiser
		4	numSubevents: The number of subevents. N = 0x01 to 0x80
		5	subeventInterval: Time from the start of one subevent to the start of the next subevent N = 0x06 to 0xFF Time = N × 1.25 ms Time Range: 7.5 ms to 318.75 ms Ignored if numSubevents = 0x01
		6	responseSlotDelay: Time from the start of one subevent to the first response slot N = 0x01 to 0xFE Time = N × 1.25 ms Time Range: 1.25 ms to 317.5 ms
		7	responseSlotSpacing: Time from the start of one response slot to the start of the next response slot N = 0x02 to 0xFF Time = N × 0.125 ms Time Range: 0.25 ms to 31.875 ms

Table 1.23: Periodic Advertising Response Timing Information data type



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 or Service Class UUIDs » [E25277_r01]
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 or Service Class UUIDs » [E25277_r01]
0x01	Length of this data
0x07	«Complete list of 128-bit Service or Service Class UUIDs » [E25277_r01]
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»

Table 2.2: Example advertising data with AD types



Value	Notes
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'

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



Value	Notes
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

2.3 ENCRYPTED ADVERTISING DATA SAMPLE DATA

2.3.1 Encrypted Advertising Data Set 1

This sample data contains an Encrypted Data data type with complete local name and AD_Appearance data types.

```
tag: Complete Local Name "Short Mini-Bus"
tag: AD_Appearance "Minibus"
```

Before Encryption

```
0F095368 6F727420 4D696E69 2D427573
03190A8C
```

Randomizer (MSO to LSO) : 0xDECA57E118

IV (MSO to LSO) : 0x46E77AB1EF007A9E

Session Key (MSO to LSO) : 0x57A9DA12D12E6E131E20612AD10A6A19

```
b0 = 4918E157 CADE9E7A 00EFB17A E7460014
b1 = 0001EA00 00000000 00000000 00000000
b2 = 0F095368 6F727420 4D696E69 2D427573
b3 = 03190A8C 00000000 00000000 00000000
```

```
x1 = C1009E37 B89726AF 303F17A6 67FFA034
x2 = BF2B64AF 43766531 EF197F2E F299B4CE
x3 = C0FAAD29 268A03DA 242ED0A1 CF3D0EA7
x4 = D65D8B4F C872FAAA B2C5A663 4FD96A55
=> MIC = D65D8B4F
```

```
a0 = 0118E157 CADE9E7A 00EFB17A E7460000
```

Data Types Specification

```
a1 = 0118E157 CADE9E7A 00EFB17A E7460001
a2 = 0118E157 CADE9E7A 00EFB17A E7460002
```

```
s0 = 6C5DE283 43EA1D81 E792D7EE 9C156BAF
s1 = 7BED8FC7 B323B308 6579AC48 524C399C
s2 = 405A1293 0F6482F5 CEF58FB1 83A3B7BA
```

```
Encrypted Payload :
    74E4DCAF DC51C728 2810C221 7F0E4CEF
    4343181F
```

```
Encrypted MIC :
    BA0069CC
```

```
Encrypted Data AD Data :
    18E157CA DE74E4DC AFDC51C7 282810C2
    217F0E4C EF434318 1FBA0069 CC
```

```
Advertising data
    1E3118E1 57CADE74 E4DCAFDC 51C72828
    10C2217F 0E4CEF43 43181FBA 0069CC
```

2.3.2 Encrypted Advertising Data Set 2

This sample data contains an Encrypted Data data type with complete local name and AD_Appearance data types, using a different randomizer value than set 1.

```
tag: Complete Local Name "Short Mini-Bus"
tag: AD_Appearance "Minibus"
```

```
Before Encryption
    0F095368 6F727420 4D696E69 2D427573
    03190A8C
```

```
Randomizer (MSO to LSO) : 0x7A6E971C8D
IV (MSO to LSO) : 0x46E77AB1EF007A9E
Session Key (MSO to LSO) : 0x57A9DA12D12E6E131E20612AD10A6A19
```

```
b0 = 498D1C97 6E7A9E7A 00EFB17A E7460014
b1 = 0001EA00 00000000 00000000 00000000
b2 = 0F095368 6F727420 4D696E69 2D427573
b3 = 03190A8C 00000000 00000000 00000000
```

```
x1 = 842739EE 53DEBA5D E1FF9BC2 508FBFB0
x2 = 4E26837E DFE59561 9727EE3E 9E9821F6
x3 = E5A79BE7 6D97E3D6 1E2B2941 832EE813
x4 = 247A22E6 1B2C78AA 9771682B A2EA292D
=> MIC = 247A22E6
```

```
a0 = 018D1C97 6E7A9E7A 00EFB17A E7460000
a1 = 018D1C97 6E7A9E7A 00EFB17A E7460001
a2 = 018D1C97 6E7A9E7A 00EFB17A E7460002
```

Data Types Specification

```
s0 = 560F67AA 41569261 73B83D23 E7158AC0
s1 = 3A4D131E 7D25FCE2 75CCE0E2 F48D85AD
s2 = FD3C1002 113DD83E 852A0F1B F686F3F1
```

Encrypted Payload :

```
35444076 125788C2 38A58E8B D9CFF0DE
FE251A8E
```

Encrypted MIC :

```
7275454C
```

Encrypted Data AD Data :

```
8D1C976E 7A354440 76125788 C238A58E
8BD9CFF0 DEFE251A 8E727545 4C
```

Advertising data

```
1E318D1C 976E7A35 44407612 5788C238
A58E8BD9 CFF0DEFE 251A8E72 75454C
```

Supplement to the Bluetooth Core Specification
Part B

**COMMON PROFILE AND
SERVICE ERROR CODES**



CONTENTS

1	Overview of Common Profile and Service Error Codes	37
1.1	Usage Descriptions	37
1.2	List of Error Codes	37
2	Common Profile and Service Error Code Descriptions	38
2.1	Out of Range (0xFF)	38
2.2	Procedure Already in Progress (0xFE)	38
2.3	Client Characteristic Configuration Descriptor Improperly Configured (0xFD)	38
2.4	Write Request Rejected (0xFC)	38



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 to 0xFF. The Common Profile and Service Error Code Descriptions Part provides an error code usage description for each failure error code.

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, which is intended when the write operation cannot be fulfilled due to permissions.

Supplement to the Bluetooth Core Specification
Part C

**SERVICES THAT MAY USE
SECURITY MODE 4 LEVEL 0**



CONTENTS

1	Services That May Use Security Mode 4 Level 0	41
1.1	Security Mode 4 Level 0 over L2CAP Connection-oriented Channels	41
1.2	Security Mode 4 Level 0 over the L2CAP Connectionless Data Channel	41



1 SERVICES THAT MAY USE SECURITY MODE 4 LEVEL 0

The following sections enumerate the services that may use Security Mode 4 Level 0 as defined in [Vol 3] Part C, Section 5.2.2.8.

[Section 1.1](#) enumerates those services that may use Security Mode 4 Level 0 over L2CAP connection oriented channels and [Section 1.2](#) enumerates those services that may 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 [Vol 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 [Vol 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)



Bluetooth SIG Proprietary and Confidential