Accessory USB-Triggered Bluetooth Pairing Specification

Release R1

Developer

1. Overview 3
1.1. Mechanism 3
1.2. Product Registration 3
2. Host Decision Logic 4
3. Accessory Requirements 4
4. HID Usages and Definitions 5
4.1. Out-of-Band Bluetooth Pairing Usage Page (0xFF44) 5
4.2. Report Descriptor Requirements 7
4.3. Examples 8
4.3.1. Report Descriptor 8
4.3.2. Report Format Pseudocode 9
5. USB Platform Descriptor 10
6. Revision History 10

1. Overview

Purpose. A HID accessory delivers what the host needs to establish or restore an LE bond over the USB wire (a trusted channel), so the host can pair or reconnect with no user-entered codes. The expected HID usages and definitions are defined in this document. **This revision is Bluetooth LE only.**

Design stance. The accessory is passive; pairing is driven by the host. The accessory advertises with **resolvable private addresses (RPAs)** and holds at most **one bond**.

1.1. Mechanism

- The host queries two sets of information over USB via **GET** Report:
  - **Address Query** — returns the accessory's current RPA.
  - **Pairing Information Query** — returns the out-of-band (OOB) pairing material.
- Additionally the host issues a **Reset Identity Command** via **SET** Report which instructs the accessory to:
  - **Disconnect** any currently connected host.
  - **Delete** any existing pairing record.
  - **Rotate** its Identity Resolving Key (IRK)
  - **Advertise** with its newly generated RPA.

1.2. Product Registration

To enable USB-triggered Bluetooth pairing in an accessory, developers must submit the information shown in the table below through the portal. After this information syncs to Apple's servers, the accessory can use the USB-triggered pairing feature.

Requirements	Size
Manufacturer company ID	2 bytes
Accessory model ID 3 bytes	3 bytes
Accessory category 1 byte	1 byte
Accessory supported mode 1 byte	1 byte

App bundle ID	< 500 bytes
Accessory Tx power	1 byte
App required for setup	1 byte
Asset for accessory	< 500 KB
Friendly name for accessory	64 bytes

2. Host Decision Logic

On plug-in the host reads the identity report and resolves the accessory's RPA locally:

Pairing State	Accessory state	Host action
Paired	Connected (to this host)	**None**
Paired	Disconnected	**Reconnect**
Unpaired	*N/A*	Issue **Reset Identity Command** → Issue **Pairing Information Query** → Pairing

3. Accessory Requirements

#	Requirement
1	The accessory must support Bluetooth Core Specification 4.2 or later, **LE,** and pair with **LE Secure Connections** in the out-of-band association model with MITM protection, a genuine (non-debug) P-256 key pair, bonding, and a 128-bit encryption key; smaller key sizes and legacy pairing must be rejected.
2	The accessory must store **at most one bond** (keys + one IRK) at a time. A new out-of-band pairing supersedes any existing bond.
3	The accessory must advertise with RPAs derived from its IRK.
4	Whenever it is bonded but disconnected the accessory must advertise with connectable advertisements.
5	On the **Address Query** read the accessory must return its current RPA.

#	Requirement
6	On the **Pairing Information Query** read the accessory must return valid OOB material (SC random + confirm over its P-256 key). Refer to *Bluetooth Core Specification v6.1, Vol 3, Part H (Security Manager Specification).*
7	Upon receiving the **Reset Identity Command**, the accessory must: disconnect any active link, delete its bond, **rotate its IRK**, and become connectable and in pairing mode with the new IRK — regardless of current link/bond state.
8	The accessory must distribute its IRK and identity address during the bonding key-distribution phase (Identity Information / Identity Address Information)
9	In the **Pairing Information Query** that follows a **Reset Identity Command**, the accessory must not return stale or invalid data. It must return its newly generated pairing information. The host will subsequently connect to the accessory and start pairing.
10	In the pairing that follows a **Pairing Information Query**, the accessory must use the random value and public key reflected in the confirm value it returned.
11	If the accessory loses its bond through a factory reset or firmware update, it must reset its identity with the same procedure defined in the **Reset Identity Command**

4. HID Usages and Definitions

4.1. Out-of-Band Bluetooth Pairing Usage Page (0xFF44)

The Out-of-Band Bluetooth Pairing Usage Page (0xFF44) is an Apple-specific vendor-defined usage page which contains the usages required to describe an accessory which supports out-of-band pairing with an Apple device.

Usage ID	Usage Name	Usage Type	Description
0000	*Reserved*		

Usage ID	Usage Name	Usage Type	Description
0001	Out-of-Band Pairing Device	CA	An application-level collection that identifies a device which supports out-of-band Bluetooth pairing.
0002-000F	*Reserved*		
0010	Bluetooth PID	SV	Bluetooth Product ID. 16-bit unsigned integer.
0011	Bluetooth VID	SV	Vendor ID. 16-bit unsigned integer.
0012	Secure Connections Random Number	SV	Random Number. Little-endian 128-bit (16-octet) value.
0013	Secure Connections Confirm Value	SV	Confirm Value. Little-endian 128-bit (16-octet) value.
0014-001F	*Reserved*		
0020	Device Address	SV	Bluetooth Device Address. Big-endian 48-bit (6-octet) value (that is, the first byte in this report field corresponds to the leftmost character in the typical bluetooth address representation XX:XX:XX:XX:XX:XX).
0021	Device Address Type	NAry	Type of Bluetooth device address.
0022	Device Address Type: Public	Sel	Public address.
0023	Device Address Type: Random	Sel	Random address.
0024-002F	*Reserved*		
0030	Transport Type	NAry	Transport type for pairing.
0031	Transport Type: BR/EDR	Sel	BR/EDR
0032	Transport Type: BLE	Sel	LE
0033-003F	*Reserved*		
0040	Accessory Type	NAry	Accessory Type
0041	Accessory Type: None	Sel	No defined accessory type.
0042	Accessory Type: Stylus	Sel	Stylus.
0043-00FF	*Reserved*		

Usage ID	Usage Name	Usage Type	Description
0100	Reset Identity	OSC*	The host asserts this control to cause an accessory to reset its identity in preparation for a fresh out-of-band pairing sequence. When the host asserts this control, a device must perform the following actions: 1. Disconnect from the currently connected Bluetooth host, if any. 2. Clear the existing Bluetooth pairing, if any. 3. Rotate its IRK 4. Generate a new RPA 5. Generate a new Secure Connection Random Number and Confirm Value. 6. Advertise, using the RPA generated in step 4. When this control is asserted, the existing Random Number, Confirm Value, or Address (that is, from before the control was asserted) are immediately considered stale. A device must never report stale values for these usages when queried by the host. **Note: the host is not required to de-assert this control to re-trigger the actions described above. A device must treat the control as though it has been de-asserted once the above actions have been completed. This is a deviation from the description of the OSC usage type in Section 3.4.1.4 of the HID Usage Tables standard.*
0101-FFFF	*Reserved*		

4.2. Report Descriptor Requirements

The top-level application collection of an out-of-band Bluetooth pairing device must have a usage ID of Out-of-Band Pairing Device.

The report descriptor must define a constant (read-only) feature item with each of the following usage IDs: Bluetooth PID, Bluetooth VID, Secure Connections Random Number, Secure Connections Confirm Value, and Device Address, as well as a constant (read-only) named-array feature item for each of the following usages: Device Address Type, Transport Type, and Accessory Type. The accessory must select exactly one entry from each named array, subject to the following additional requirements:

The Device Address Type array must select the Device Address Type: Random entry (usage ID 0xFF440023). Accessories with public addresses are not supported for out-of-band pairing.

The Transport Type array must select the Transport Type: BLE entry (usage ID 0xFF440032). Bluetooth BR/EDR accessories are not supported for out-of-band pairing.

The report descriptor must also define an output item with the Reset Identity usage ID.

4.3. Examples

4.3.1. Report Descriptor

```
0x06, 0x44, 0xFF,           // Usage Page (0xFF44 - Apple Vendor Out of Band Pairing)
0x09, 0x01,                 // Usage (0x01 - Out of Band Pairing Device)
0xA1, 0x01,                 // Collection (Application)
0x85, 0x01,                 //   Report ID (1)
0x15, 0x00,                 //   Logical Minimum (0)
0x27, 0xFF, 0xFF, 0x00, 0x00,  //   Logical Maximum (65535)
0x75, 0x10,                 //   Report Size (16)
0x95, 0x01,                 //   Report Count (1)
0x09, 0x10,                 //   Usage (0x10 - Bluetooth PID)
0xB1, 0x03,                 //   Feature (Const, Var, Abs)
0x09, 0x11,                 //   Usage (0x11 - Bluetooth VID)
0xB1, 0x03,                 //   Feature (Const, Var, Abs)
0x15, 0x00,                 //   Logical Minimum (0)
0x26, 0xFF, 0x00,           //   Logical Maximum (255)
0x75, 0x08,                 //   Report Size (8)
0x95, 0x10,                 //   Report Count (16)
0x09, 0x12,                 //   Usage (0x12 - Secure Connections Random Number)
0xB1, 0x03,                 //   Feature (Const, Var, Abs)
0x09, 0x13,                 //   Usage (0x13 - Secure Connections Confirm Value)
0xB1, 0x03,                 //   Feature (Const, Var, Abs)
0x75, 0x08,                 //   Report Size (8)
0x95, 0x06,                 //   Report Count (6)
0x09, 0x20,                 //   Usage (0x20 - Bluetooth Device Address)
0xB1, 0x03,                 //   Feature (Const, Var, Abs)
0x09, 0x21,                 //   Usage (0x21 - Device Address Type)
0xA1, 0x04,                 //   Collection (Named Array)
0x15, 0x00,                 //     Logical Minimum (0)
0x25, 0x01,                 //     Logical Maximum (1)
0x75, 0x01,                 //     Report Size (1)
0x95, 0x01,                 //     Report Count (1)
0x09, 0x22,                 //     Usage (0x22 - Device Address Type: Public)
0x09, 0x23,                 //     Usage (0x23 - Device Address Type: Random)
0xB1, 0x01,                 //     Feature (Const, Array, Abs)
0xC0,                       //   End Collection
0x09, 0x30,                 //   Usage (0x30 - Transport Type)
0xA1, 0x04,                 //   Collection (Named Array)
0x15, 0x00,                 //     Logical Minimum (0)
0x25, 0x01,                 //     Logical Maximum (1)
0x75, 0x01,                 //     Report Size (1)
0x95, 0x01,                 //     Report Count (1)
0x09, 0x31,                 //     Usage (0x31 - Transport Type: BR/EDR)
0x09, 0x32,                 //     Usage (0x32 - Transport Type: LE)
```

```
0xB1, 0x01,                 //    Feature (Const, Array, Abs)
0xC0,                       //  End Collection
0x09, 0x40,                 //  Usage (0x40 - Accessory Type)
0xA1, 0x04,                 //  Collection (Named Array)
0x15, 0x00,                 //    Logical Minimum (0)
0x25, 0x01,                 //    Logical Maximum (1)
0x75, 0x01,                 //    Report Size (1)
0x95, 0x01,                 //    Report Count (1)
0x09, 0x41,                 //    Usage (0x41 - Accessory Type: None)
0x09, 0x42,                 //    Usage (0x42 - Accessory Type: Stylus)
0xB1, 0x01,                 //    Feature (Const, Array, Abs)
0xC0,                       //  End Collection
0x75, 0x05,                 //  Report Size (5)
0xB1, 0x03,                 //  Feature (Const, Var, Abs) (Padding)
0x15, 0x00,                 //  Logical Minimum (0)
0x25, 0x01,                 //  Logical Maximum (1)
0x75, 0x01,                 //  Report Size (1)
0x95, 0x01,                 //  Report Count (1)
0x0A, 0x00, 0x01,           //  Usage (0x100 - Reset Identity)
0x91, 0x06,                 //  Output (Data, Var, Rel)
0x75, 0x07,                 //  Report Size (7)
0x91, 0x03,                 //  Output (Const, Var, Abs) (Padding)
0xC0,                       // End Collection
```

4.3.2. Report Format Pseudocode

The following C types represent the format of the report descriptor in section 4.3.1:

```
typedef enum : uint8_t {
    kDeviceAddressTypePublic = 0,
    kDeviceAddressTypeRandom = 1,
} OOBPairingDeviceAddressType;

typedef enum : uint8_t {
    kTransportTypeBREDR = 0,
    kTransportTypeBLE   = 1,
} OOBPairingTransportType;

typedef enum : uint8_t {
    kAccessoryTypeNone   = 0,
    kAccessoryTypeStylus = 1,
} OOBPairingAccessoryType;

typedef struct __attribute__((packed)) {
    uint8_t report_id;
    uint16_t bluetooth_pid;
    uint16_t bluetooth_vid;
    uint8_t secure_connections_random_number[16];
    uint8_t secure_connections_confirm_value[16];
    uint8_t device_address[6];
    OOBPairingDeviceAddressType device_address_type : 1;
    OOBPairingTransportType transport_type : 1;
    OOBPairingAccessoryType accessory_type : 1;
    uint8_t padding : 5;
} OOBPairingFeature01;

typedef struct __attribute__((packed)) {
    uint8_t report_id;
    uint8_t reset_identity : 1;
    uint8_t padding : 7;
```

} OOBPairingOutput01;

5. USB Platform Descriptor

To indicate to Apple devices that vendor-defined usage IDs in its report descriptor should be interpreted according to Apple definitions, accessories must define a Platform Descriptor with the following PlatformCapabilityUUID (see Section 9.6.2.4 of the USB 3.2 specification):

PlatformCapabilityUUID
D463ECFD-A39B-49ED-AE94-B8C84D44BA11

6. Revision History

Version	Date	Notes
R1	2026-10-09	Release R1

Apple Inc.

Copyright © 2026 Apple Inc.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer or device for the intended use only and to print copies of documentation for the intended use provided that the documentation contains Apple's copyright notice.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to be used in the development of solutions for Apple-branded products.

Apple Inc.

One Apple Park Way

Cupertino, CA 95014

408-996-1010

Apple and the Apple Logo are trademarks of Apple Inc., registered in the U.S. and other countries.

APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT, ERROR OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

Some jurisdictions do not allow the exclusion of implied warranties or liability, so the above exclusion may not apply to you.