# Accessory Proximity-Triggered Pairing Specification

## Release R1.1

Developer

2026-09-01 | Copyright © 2026 Apple Inc. All Rights Reserved.

Accessory Proximity-Triggered Pairing Specification 1
Release R1.1 1
1. Introduction 5
1.1. Terminology 6
2. Core concepts 7
2.1. Overview 7
2.2. Transport 7
2.3. Operation 7
2.4. Accessory onboarding 7
2.5. Advertising payload 8
2.6. Supported modes 9
2.7. Subtype 10
2.8. Accessory name 10
2.9. Pairing flow 10
2.9.1. Bluetooth LE (app required to pair) 11
2.9.2. Bluetooth LE (includes system services) 13
2.9.3. Dual-mode device (app required to pair) 15
2.9.4. Dual-mode device (includes system services) 16
2.9.5. Accessory supports Classic only 17
3. Requirements 18
3.1. Overview 18
3.2. General 18
4. Bluetooth requirements 18
4.1. Overview 18
4.2. Bluetooth advertising 18
4.3. Bluetooth connection 19
4.4. Bluetooth host 19

2026-09-01 | Copyright © 2026 Apple Inc. All Rights Reserved.

4.4.1.MTU size 19
4.4.2.Link-encryption key 19
4.4.3.Handling concurrent operations 19
4.4.4.Timeout 19
5. Product registration and self-certification 20
5.1. Requirements 20
5.2. Testing 20
6. iOS 26.5 release 22
6.1. Introduction 22
6.2. Release notes 22
7. Self-certification tool 25
7.1. Scope 25
7.2. Background 25
7.3. Functional requirements 25
7.4. Input and assumptions 25
7.5. Data collection 26
7.6. Path loss calculation 26
7.6.1.Sampling duration 26
7.7. Controlled RF chamber (recommended) 26
7.8.Semicontrolled indoor (office setup) 26
7.9. Edge cases and limitations 27
7.10.Security and privacy 27
7.11.Environmental factors 27
7.12.Device-specific factors 27
7.13.Measurement strategy 27
7.14.Interpretation pitfalls 28
7.15.Test setup instructions and data collection 29

2026-09-01 | Copyright © 2026 Apple Inc. All Rights Reserved.

7.16. Enabling the data-collection tool 29
8. Asset details 32
9. Revision history 33

2026-09-01 | Copyright © 2026 Apple Inc. All Rights Reserved.

# 1. Introduction

This document outlines the process of setting up an accessory with Apple devices when in proximity to each other. This enables seamless setup and the assisted download of the associated application from the App Store.

You can develop and test proximity pairing on devices in any region using an Apple-provided provisioning profile. However, in production, people using your app must have an Apple Account registered in the European Union (EU), and their device must be located within the EU.

2026-09-01 | Copyright © 2026 Apple Inc. All Rights Reserved.

## 1.1. Terminology

Throughout this document, these terms have specific meanings:

- The term *Apple device* refers to an an iPhone or iPad running iOS or iPadOS, respectively.
- The term *accessory* refers to any product intended to interface with a device through the means that this specification describes.
- The term *Apple Account* refers to the account system Apple uses to provide access to Apple devices and services, including the App Store, iCloud, and related features.
- The term *SIG* refers to the Bluetooth Special Interest Group (SIG), a network of member organizations that are the caretakers and innovators of Bluetooth® technology.

2026-09-01 | Copyright © 2026 Apple Inc. All Rights Reserved.

# 2. Core concepts

## 2.1. Overview

The *Accessory Proximity-Triggered Pairing Specification* outlines the process of setting up an accessory with Apple devices when they are in proximity to each other. This enables seamless setup and the assisted download of the associated application from the App Store. If the accessory is compatible with both iPhone and iPad, ensure that the companion app supports both iOS and iPadOS.

## 2.2. Transport

The Accessory Proximity Pairing protocol uses Bluetooth Low Energy (BLE) as its primary mode of communication with Apple devices. If the accessory primarily supports Bluetooth Classic profiles, proximity discovery is based on the BLE advertisement alone. If the accessory can't support BLE advertising, then proximity setup isn't possible.

## 2.3. Operation

When an accessory is in pairing or setup mode, it advertises via a Bluetooth Low Energy (BLE) beacon with a specific payload to indicate it's in this mode. If an Apple device is close enough to the accessory, it picks up this beacon and prompts the user to perform setup.

## 2.4. Accessory onboarding

### 2.4.1. Accessory

An *accessory* is the device that implements the Accessory Proximity Pairing protocol to provide a seamless setup experience.

### 2.4.2. Apple Server

Apple servers store information about the accessory, such as the product type, the product image, the associated app bundle, and the supported services that are available when the accessory has been registered with the program.

### 2.4.3. Company ID

In Bluetooth Low Energy (BLE), the company ID is a 16-bit value that the Bluetooth Special Interest Group (SIG) assigns to a specific company or organization. The manufacturer-specific data element within BLE advertising packets uses this identifier.

2026-09-01 | Copyright © 2026 Apple Inc. All Rights Reserved.

To avoid conflicts and ensure interoperability between different Bluetooth devices, the Bluetooth SIG typically assigns a unique company ID to each company or organization. Having unique identifiers allows devices to recognize and interpret the manufacturer-specific data correctly.

### 2.4.4. Accessory model ID

A 24-bit model ID is required at registration. The manufacturer determines this value — it can be any 24-bit number. Use it to distinguish between models within a product family, such as by color, size, or other variation.

### 2.4.5. RSSI offset

For this release, the pairing card appears at 20 centimeters, consistent with all other Apple accessories. The self-certification tool calculates this threshold. The received signal strength indicator (RSSI) offset byte is the two's complement representation of the RSSI value encoded as an unsigned byte.

For example, if the RSSI on the phone for the accessory is -55 dBm, use that as the last byte of the payload: 256 – 55 = 201, or 0xC9 in hexadecimal.

For –45 dBm: 256 – 45 = 211, or 0xD3. If the RSSI is too low, then the distance at which the pairing card shows up on the phone increases.

## 2.5. Advertising payload

This document uses BLE(Bluetooth Low Energy) advertisement packets for discovery. The Bluetooth specifications authoritatively define the format of these packets. To simplify implementations, this section provides a summary of the specific BLE packet formats that this document uses.

Bluetooth data link layer packets deliver Bluetooth Low Energy (BLE) advertisements, using the BLE Advertising Channel PDU. This PDU has the following format:

Table 2-1 Payload for Pairing State

| Byte | Value | Description |
|---|---|---|
| 0 | 10 (Length) | Length of the ProximityPairing payload |
| 1 | 0x16 | 16-bit UUID service data type |
| 2-3 | 0xFE25 | 16-bit UUID for ProximityPairing service |
| 4 | Advertising Subtype | See section 2.7 for Subtypes |
| 5-6 | CompanyID | Section 2.4.3 for CompanyID |
| 7-9 | Accessory ModelID | Section 2.4.4 for Accessory ModelID |
| 10 | RSSI Offset | RSSI Offset for the accessory |

2026-09-01 | Copyright © 2026 Apple Inc. All Rights Reserved.

## 2.6. Supported modes

At registration, provide a 1-byte supported mode value. This value helps determine the capabilities of the accessory and provide the appropriate user experience flow. For more information on the supported modes, see below:

### 2.6.1. Bluetooth Low Energy only (app required to complete setup)

The accessory supports a custom service that requires the manufacturer's application to complete the setup and enable the functionality of the accessory. This could also be used if the accessory is advertising a standard Bluetooth service that the Bluetooth SIG defines. Use this mode only when the accessory exclusively supports Bluetooth LE for connectivity and doesn't support Bluetooth pairing. If the accessory is a dual-mode device, see section 2.6.3.

### 2.6.2. Bluetooth Low Energy only (system services)

The accessory supports a built-in system service that Apple specifies, such as the Human Interface Device (HID) service, or the Apple Notification Center Service (ANCS). After completing the Bluetooth pairing and authentication, the user will be guided through the setup for the services that the accessory supports. If there is a  required companion app, the user receives a prompt to download the application along with the setup process.

### 2.6.3. Bluetooth Low Energy and Bluetooth Classic (dual mode)

A dual-mode accessory that supports both Bluetooth LE and Bluetooth Classic should provide this support at the time of registration. If the accessory supports a standard Bluetooth Classic service supported by iOS, then it will be automatically configured and available for the user. For the custom Bluetooth LE services, the companion application will be presented to the user to download and continue to the setup after authorization.

### 2.6.4. Bluetooth Classic and Bluetooth Low Energy advertising only

An accessory that supports only Bluetooth Classic services but has the capability to advertise over Bluetooth Low Energy should enable this mode for advertising. When this mode is enabled, iOS connects directly over Bluetooth Classic and sets up supported services.

The table below provides details of the supported-services byte that the accessory needs to advertise, to provide the best setup experience based on the supported services:

2026-09-01 | Copyright © 2026 Apple Inc. All Rights Reserved.

Table 2-2 Supported Modes

| Description |
| --- |
| Supported Modes:<br>00: Bluetooth LE (App Required to pair)<br>01: Bluetooth LE (Includes System services)<br>02: Dual mode device (App Required to pair)<br>03: Dual mode device (Includes System services)<br>04: Bluetooth Classic + Bluetooth Low Energy advertising only<br>05 and beyond: Reserved for future use |

## 2.7. Subtype

This document defines the following subtypes. Subtypes not defined by this document are reserved for definition by other documents.

Table 2-3 — Subtype Definitions

| SubType | | Description |
| --- | --- | --- |
| **“0x30”** | **Accessory Setup** | Advertised for Accessory setup |
| **Others** | **Reserved** | Reserved for future use |

During setup mode, the accessory must use subtype 0x30 in the advertising payload for Apple to process the advertisement and start to validate it to show the proximity card for the user. All other subtypes are reserved for future use at this time.

## 2.8. Accessory name

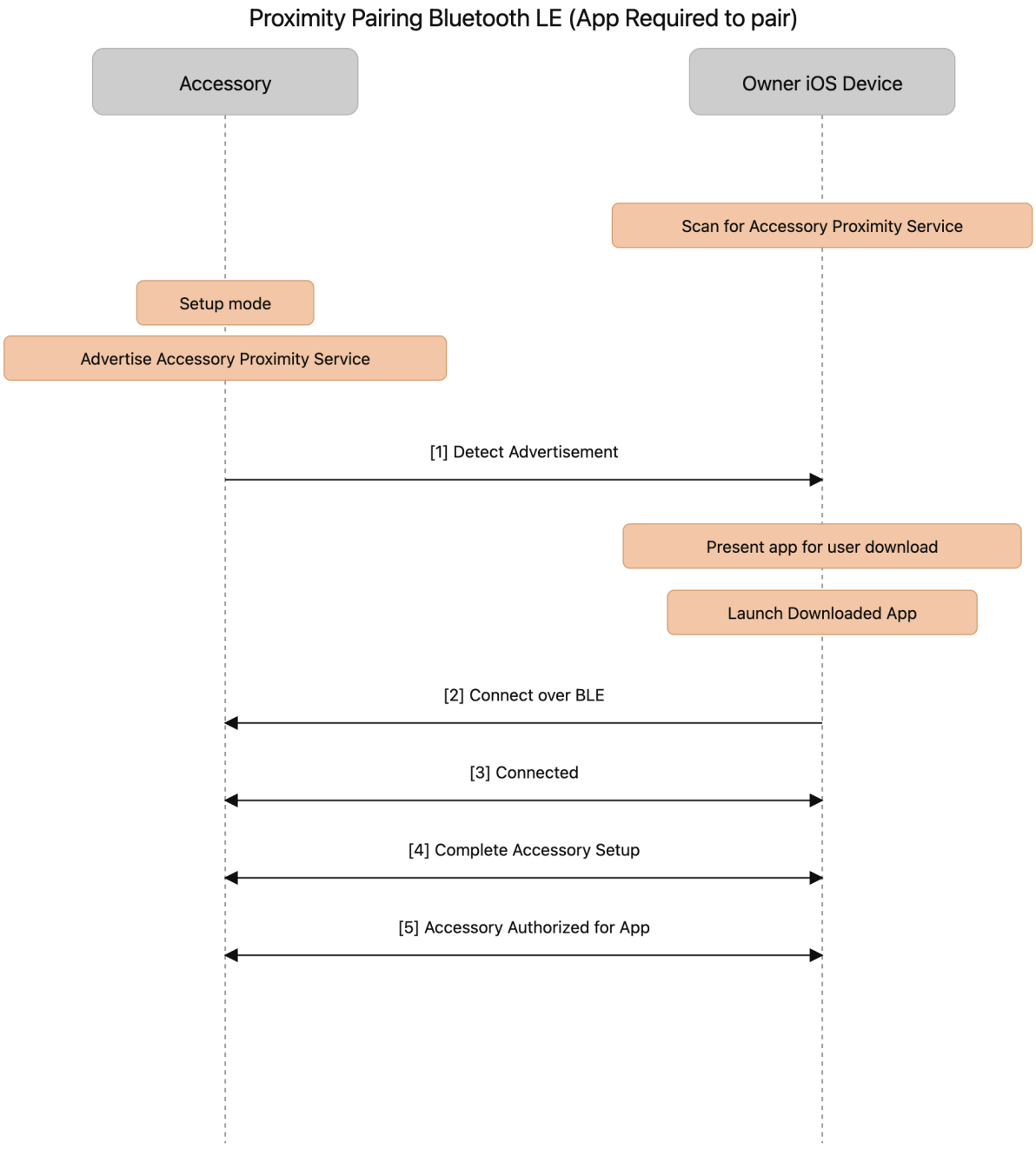
Accessory manufacturers add an over-the-air (OTA) name in the scan response. Before the proximity card is displayed, the Bluetooth stack on iOS performs an active scan and retrieves the name to add to the OTA field of the proximity card. Additionally, the manufacturer should provide a friendly name at the time of registration, which serves as the title of the proximity card. For more information on the friendly name, see section 5.

## 2.9. Pairing flow

This section describes the pairing flow for various modes the accessory supports.

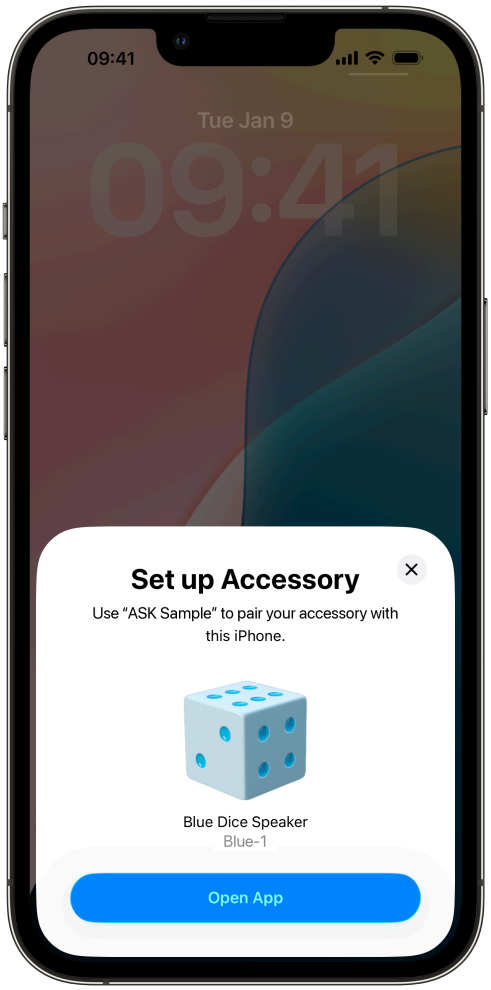
2026-09-01 | Copyright © 2026 Apple Inc. All Rights Reserved.

## 2.9.1. Bluetooth LE (app required to pair)



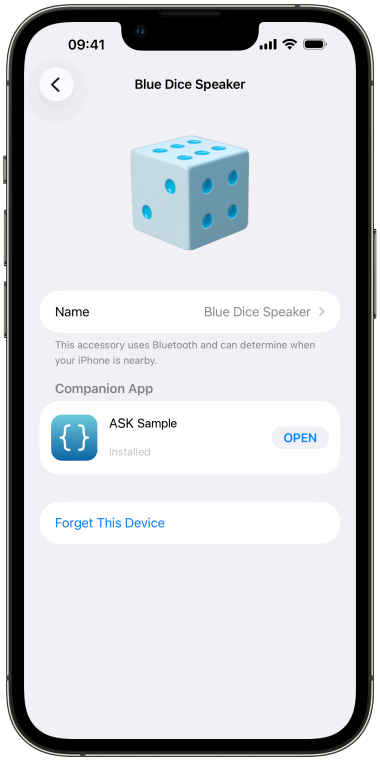

When the accessory doesn't support Bluetooth pairing and the application controls the setup, follow this flow for the optimal user experience.

2026-09-01 | Copyright © 2026 Apple Inc. All Rights Reserved.



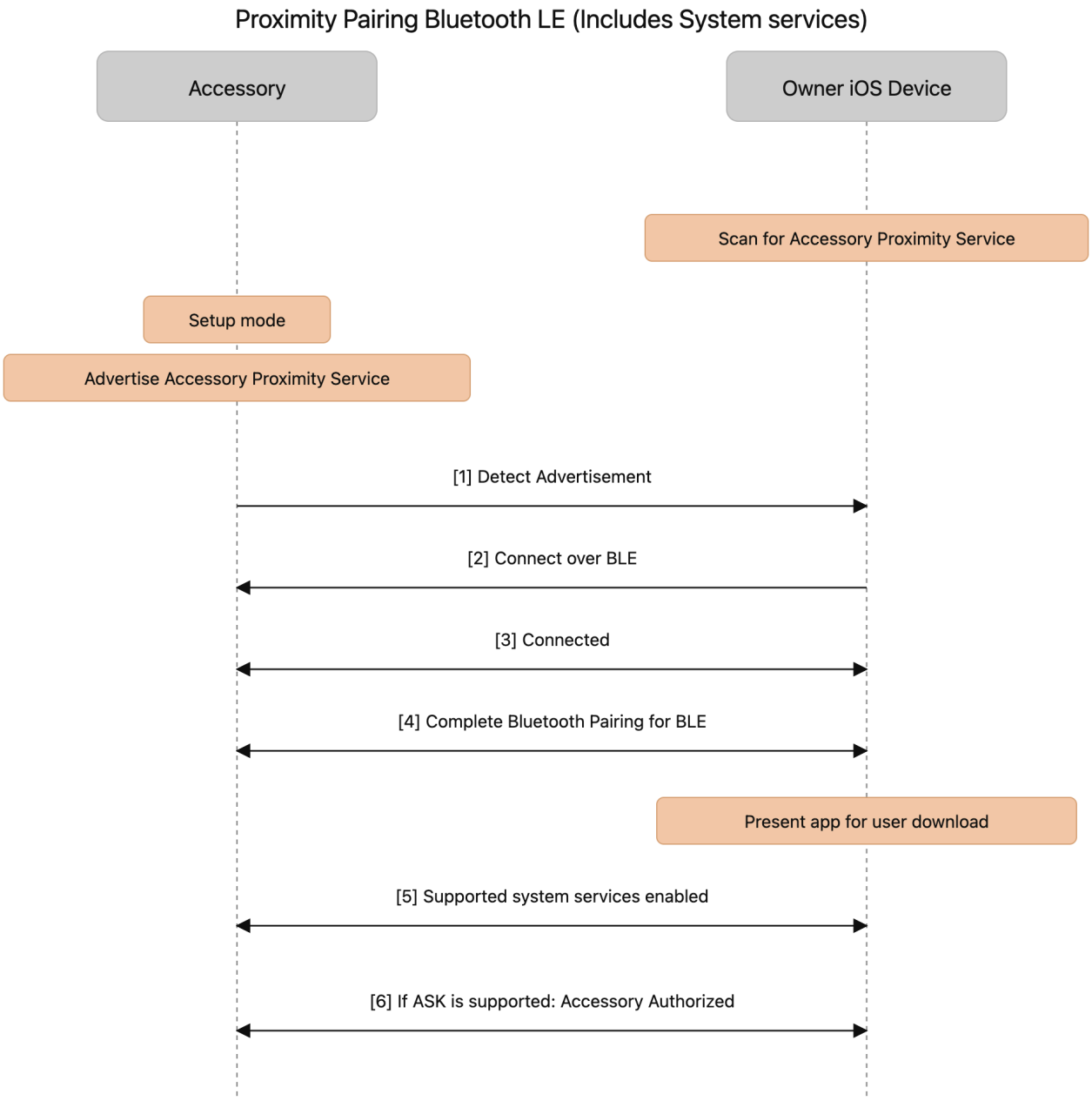






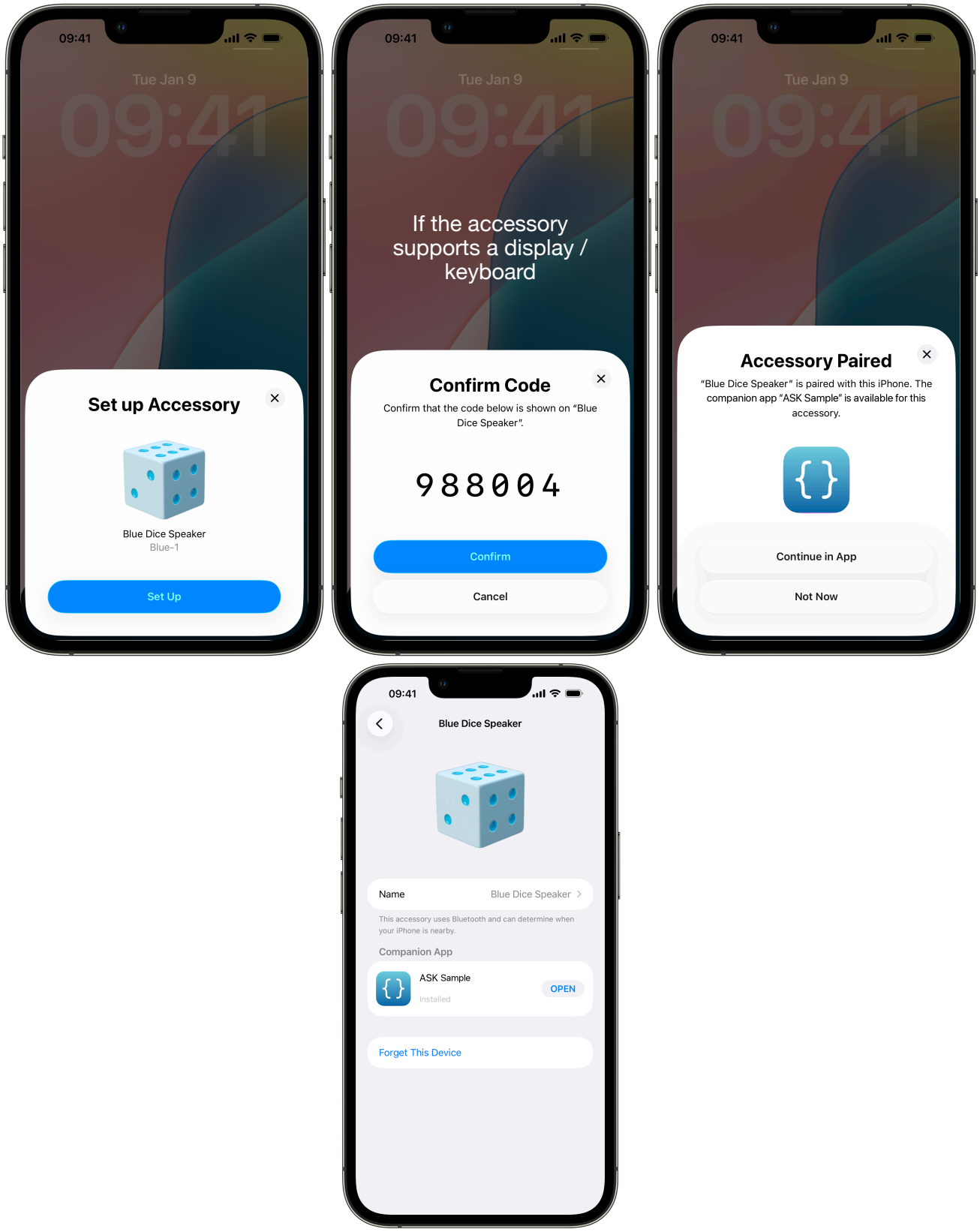

2026-09-01 | Copyright © 2026 Apple Inc. All Rights Reserved.

## 2.9.2. Bluetooth LE (includes system services)



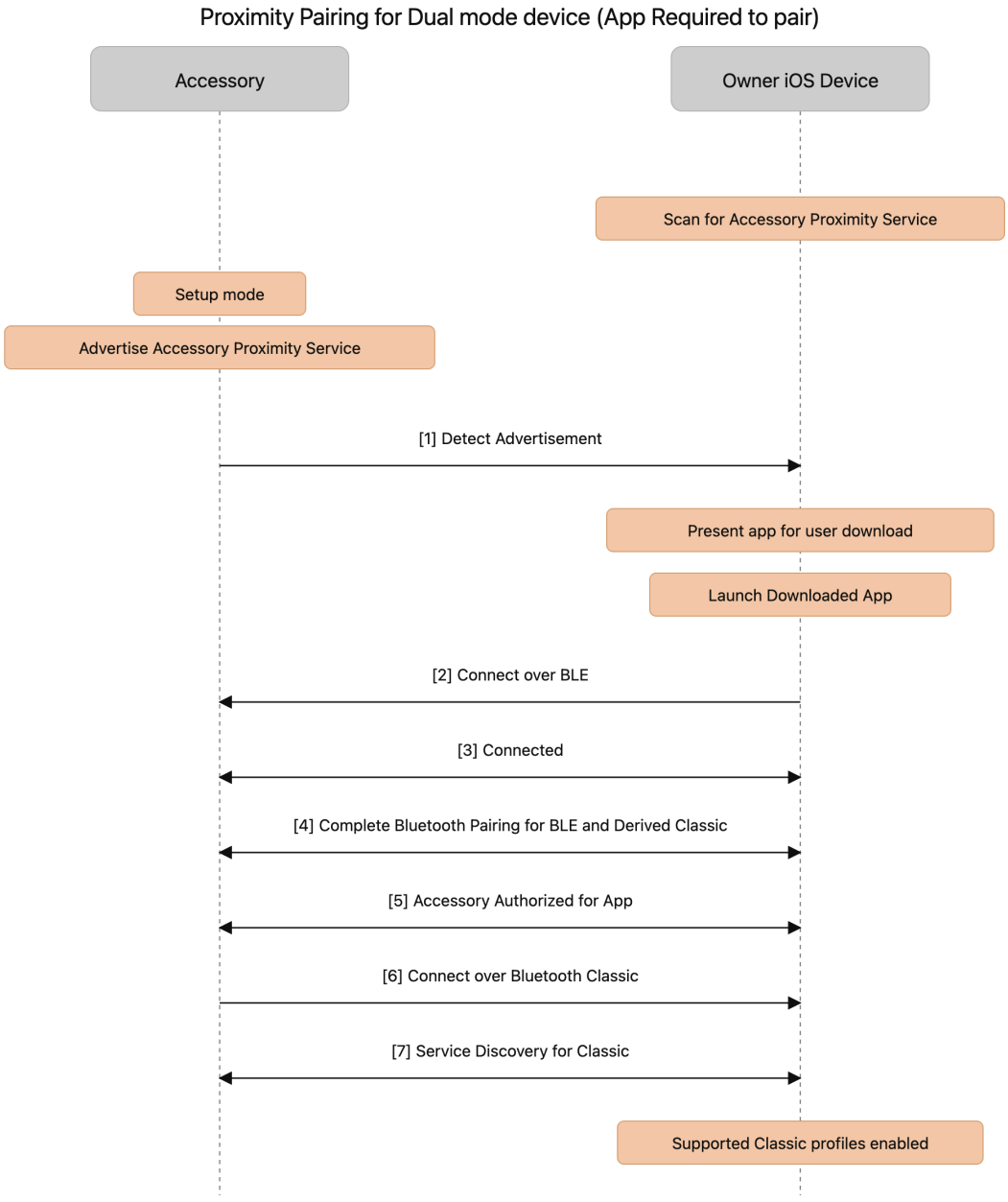

2026-09-01 | Copyright © 2026 Apple Inc. All Rights Reserved.

When the accessory supports Bluetooth pairing, follow this flow for the optimal user experience.




2026-09-01 | Copyright © 2026 Apple Inc. All Rights Reserved.

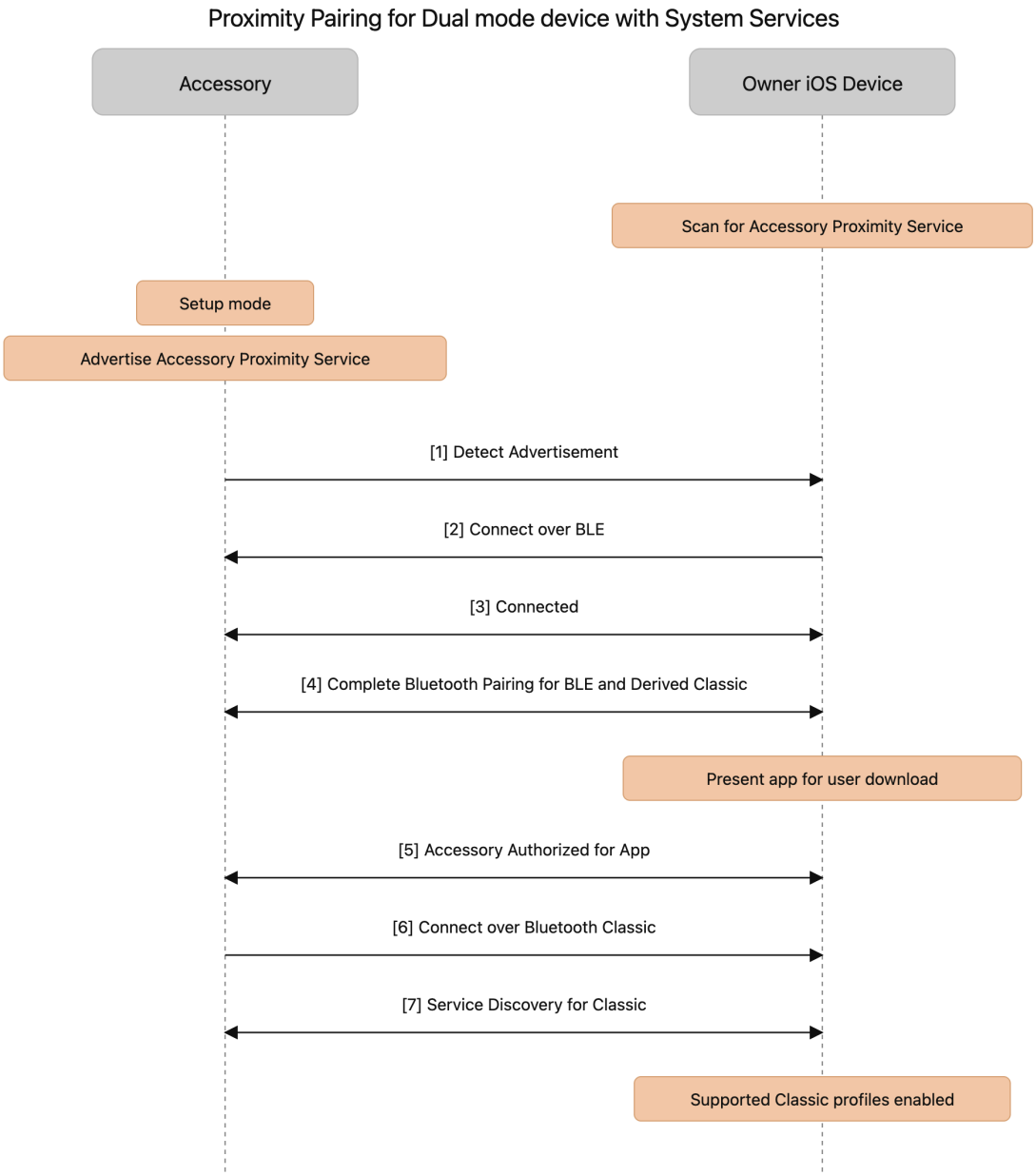
## 2.9.3. Dual-mode device (app required to pair)




This mode follows the same flow as BLE-App required. If the accessory supports GATT over Classic, then Bluetooth pairing is required. Use this flow only if setup includes custom modes that require specific instructions.

2026-09-01 | Copyright © 2026 Apple Inc. All Rights Reserved.

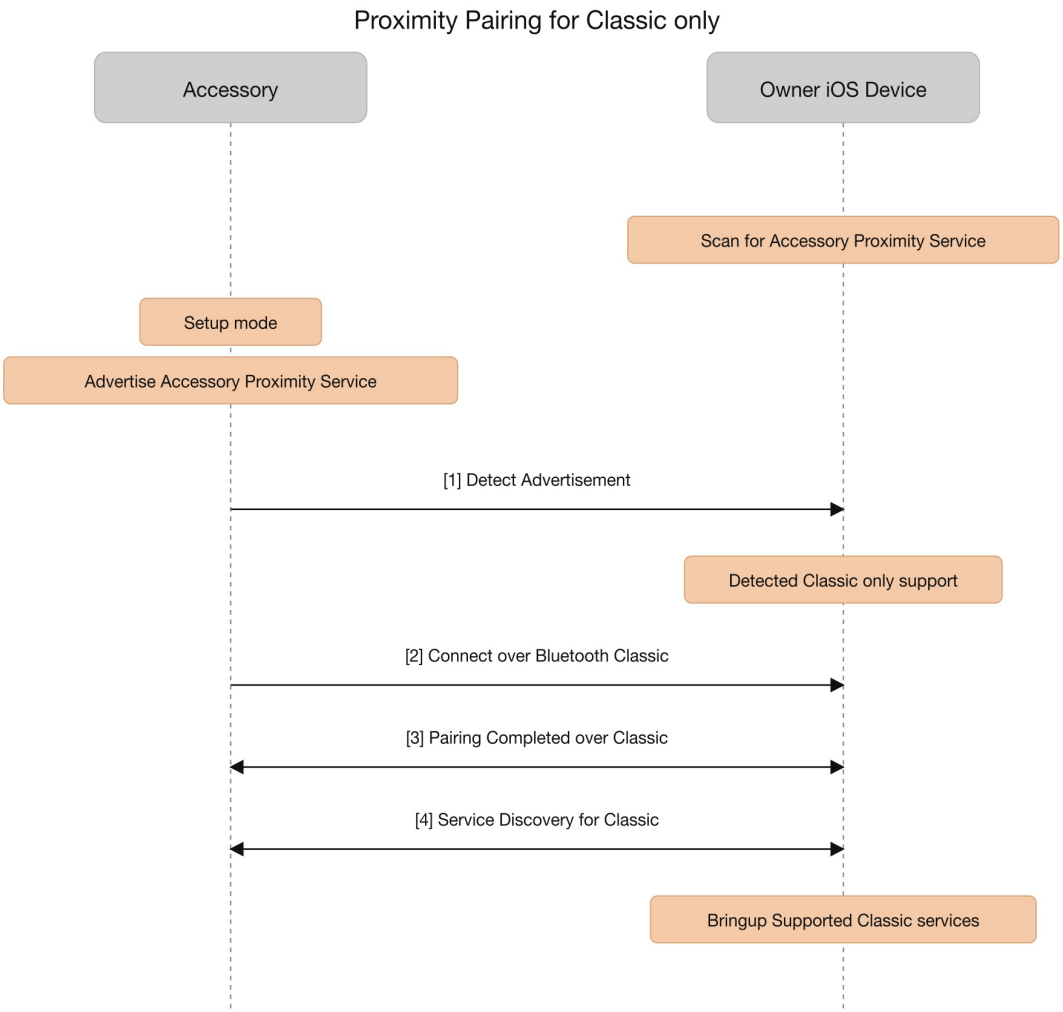
## 2.9.4. Dual-mode device (includes system services)




When the accessory supports Bluetooth pairing, follow this flow for the optimal user experience.

2026-09-01 | Copyright © 2026 Apple Inc. All Rights Reserved.

## 2.9.5. Accessory supports Classic only




When the accessory doesn’t support BLE connections and only supports Bluetooth Classic pairing, follow this flow for the optimal user experience. No BLE functionality is enabled and connections aren’t possible.

2026-09-01 | Copyright © 2026 Apple Inc. All Rights Reserved.

# 3. Requirements

## 3.1. Overview

Accessories that support the Accessory Proximity Pairing protocol must conform to the requirements this chapter lists, along with any feature-specific requirements in other chapters.

## 3.2. General

- If an accessory is powered on and isn't configured, it must enter setup mode automatically.
- If an accessory has been in setup mode for at least 2 minutes, it should exit setup mode after this time or on successful pairing.
- If the user resets an accessory, it should enter setup mode automatically.

# 4. Bluetooth requirements

## 4.1. Overview

Bluetooth Low Energy (LE) serves as the wireless transport for all communication between Apple products and accessories. The Bluetooth LE transmit power level of the accessory shall be fixed at ≥ +4dBm. The transmit power level is the conducted transmit power.

## 4.2. Bluetooth advertising

The accessory should advertise the Accessory Proximity Pairing payload when in setup mode.

The interval between advertisements should be no larger than 100 milliseconds. A fast rate allows the iOS device to quickly find the accessory, even when scanning in low-power mode. Ideally, the advertising interval for setup could be fast for the first few seconds on entering setup mode and then drop down to a lower interval.

When an accessory is in setup mode, it shall advertise using the public address or a random static address and shall not rotate the advertisement for the duration of the setup process. After pairing, the accessory should switch to a resolvable address.

2026-09-01 | Copyright © 2026 Apple Inc. All Rights Reserved.

## 4.3. Bluetooth connection

The accessory must support at least two simultaneous connections in a peripheral role.

The connection interval of the Bluetooth LE link between the Apple device and accessory depends on the type of user interaction. An Apple device typically selects a connection interval in multiples of 15 ms. The accessory shall support a connection interval that is a multiple of 15 ms.

## 4.4. Bluetooth host

### 4.4.1. MTU size

The accessory shall select an MTU size that is equal to or greater than the MTU request from the Apple device.

### 4.4.2. Link-encryption key

The accessory pairs to the Apple device using the supported Bluetooth pairing schemes (Numeric Comparison, Passkey Entry, or Just Works pairing). To encrypt the Bluetooth LE link on every subsequent connection, the Bluetooth LTK is used for encryption.

### 4.4.3. Handling concurrent operations

An app on the Apple device may interact with the accessory over GATT or, if supported, connection-oriented L2CAP channels. Apple devices may connect and perform GATT operations independently from other interactions with the accessory.

### 4.4.4. Timeout

Unless otherwise specified, the accessory must respond to all commands within 30 seconds.

2026-09-01 | Copyright © 2026 Apple Inc. All Rights Reserved.

# 5. Product registration and self-certification

## 5.1. Requirements

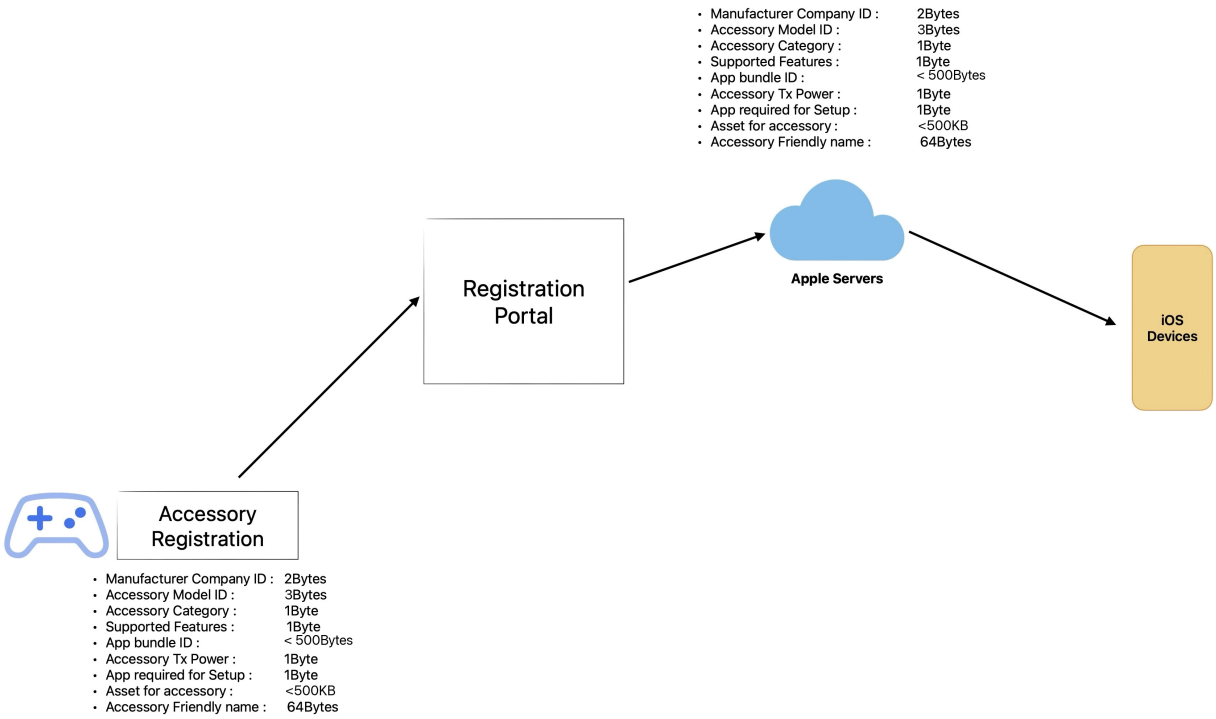
To enable proximity pairing in an accessory, developers must submit the information shown in the table below and complete self-certification through the portal. After self-certification is complete, this information syncs to Apple's servers, and the accessory can use the proximity-pairing feature.

| Requirements | Size |
|---|---|
| Manufacturer company ID | 2 bytes |
| Accessory model ID | 3 bytes |
| Accessory category | 1 byte |
| Accessory supported mode | 1 byte |
| App bundle ID | < 500 bytes |
| Accessory Tx power | 1 byte |
| App required for setup | 1 byte |
| Asset for accessory | < 500KB |
| Friendly name for accessory | 64 bytes |

## 5.2. Testing

Using a developer profile, developers can test the accessory with iOS devices. This profile also enables the tool for self-certification inside the Developer settings. The developer profile can be downloaded here.

2026-09-01 | Copyright © 2026 Apple Inc. All Rights Reserved.




2026-09-01 | Copyright © 2026 Apple Inc. All Rights Reserved.

# 6. iOS 26.5 release

## 6.1. Introduction

In iOS 26.5, proximity pairing is available for manufacturers to experiment with, iterate on the user experience, and prepare products for release.

## 6.2. Release notes

When a developer registers a product via the registration portal, the accessory details are sourced from that portal. For testing, the accessory details are taken from the installed app instead.

This release includes a test company ID that the accessory needs to advertise. The test company ID is `0xFF20`. Ensure that the accessory uses this for the company ID field in the payload. In the `Info.plist` file of your application, use the following key for the system to match the payload. Ensure that the `Info.plist` file has `NSAccessorySetupSupports` and Bluetooth so it adopts AccessorySetupKit (https://developer.apple.com/documentation/accessorysetupkit).

<key>NSAccessorySetupProximityServices</key>

<array>

</array>

The array can include the following fields:

`ModelID`, `CompanyID`, `Image`, `BundleID`, `AccessoryName`

`ModelID`: Any 24-bit ID that the manufacturer determines. Ensure the same model ID is advertised in the `modelID` field of the payload.

`CompanyID`: The developer's chosen company ID, used in the advertisement. Provide this in `Info.plist`.

`BundleID`: The bundle ID of the application that AccessorySetupKit authorizes.

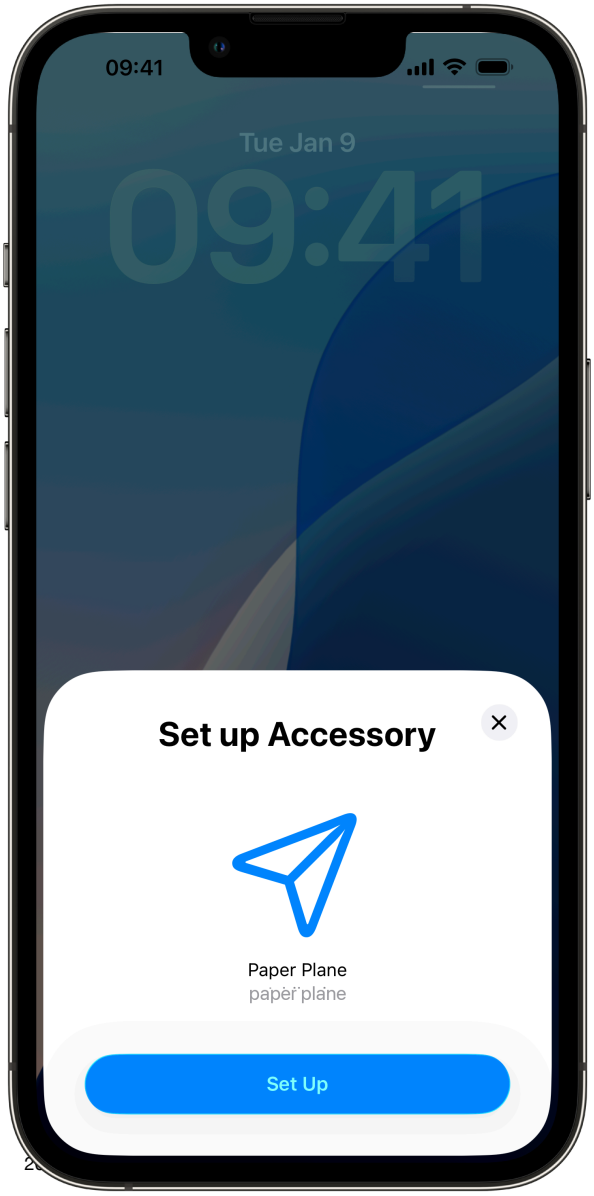
`Image`: An SF Symbol that best matches your accessory and appears on the pairing card when the accessory is discovered by the system.

`AccessoryName`: A friendly name that appears as the title of the pairing card.

2026-09-01 | Copyright © 2026 Apple Inc. All Rights Reserved.

Here is a sample of the `Info.plist` entry:

```
<key>NSAccessorySetupProximityServices</key>
        <array>
                <dict>
                        <key>ModelID</key>
                        <string>002005</string>
                        <key>CompanyID</key>
                        <string>FF20</string>
                        <key>Image</key>
                        <string>paperplane</string>
                        <key>BundleID</key>
                        <string>com.example.apple-samplecode.ASKSample</string>
                        <key>AccessoryName</key>
                        <string>Paper Plane</string>
                </dict>
        </array>
```



Pairing Card Display:

Does the app get authorized to get access to the accessory that just got paired?

Yes, instantiate an `ASAccessorySession`; if the accessory is paired and the `info.plist` has the right `bundleID`, then as soon as the app is launched there will be an event and the app will get the `accessoryAdded` callback with the paired `ASAccessory`.

```
private var session = ASAccessorySession()
self.session.activate(on: DispatchQueue.main, eventHandler: handleSessionEvent(event:))

private func handleSessionEvent(event: ASAccessoryEvent) {
    switch event.eventType {
    case .accessoryAdded, .accessoryChanged:
      guard let pairedAccessory = event.accessory
```

Below is a sample advertisement report on iOS that triggers the pairing card as captured on a sniffer:

Data: 0A 16 ***25 FE*** 30 FF 20 00 20 05 D3

UUID: 0xFE25
SubType: 0x30
CompanyID: 0xFF20
ModelID: 0x002005
RSSI Offset: 0xD3

2026-09-01 | Copyright © 2026 Apple Inc. All Rights Reserved.

# 7. Self-certification tool

This section outlines the method to generate and store Bluetooth path loss data between third-party accessories and Apple devices using Core Bluetooth APIs in the iOS development environment. Use the specification for evaluating RF performance and determining whether the accessory is within the bubble to generate the pairing card.

## 7.1. Scope

- Measuring RSSI and extracting Tx Power from BLE advertisements
- Computing path loss on iOS and/or iPadOS
- Recommendations for environments to collect repeatable and meaningful data

## 7.2. Background

Bluetooth Low Energy (BLE) accessories communicate with Apple iOS devices using advertisements and GATT profiles. Core Bluetooth exposes limited RF parameters; however, it's possible to estimate path loss using the RSSI and transmitted power (Tx Power), if available.

The Core Bluetooth framework allows access to RSSI via discovery events, and optional access to Tx Power through the `CBAdvertisementDataTxPowerLevelKey` if the advertiser includes it.

## 7.3. Functional requirements

- Scan for BLE accessories.
- Extract RSSI and Tx Power.
- Compute path loss.
- Store results locally and share.

## 7.4. Input and assumptions

- The BLE accessory is advertising.
- The accessory includes Tx Power in its advertisement packet.
- RSSI can be sampled during scanning.
- The iOS/iPadOS app is running in the **foreground**.
- The **Tx Power** included in advertisements may have an **inherent tolerance** (e.g., ±2 dB) due to hardware calibration limits on the accessory. This must be accounted for when interpreting path loss. If the accessory manufacturer specifies Tx Power accuracy or calibration offset, it

2026-09-01 | Copyright © 2026 Apple Inc. All Rights Reserved.

should be integrated into the computation model or reported alongside the result for transparency.
- The advertisement interval should be < 100 ms.

## 7.5. Data collection

- Install the provided Accessory Proximity-Triggered Pairing Test Profile.
- The app uses `CBCentralManager` to scan peripherals.
- The accessory advertises with the advertising payload mentioned in section 2.5, with the Byte 10 being the **Tx power of accessory instead of RSSI offset.**
- Capture `**RSSI**` and `**TxPower**`.
- **Note on Tx Power tolerance:**
  If the BLE accessory's Tx Power has a known tolerance (e.g., ±2 dB), factor it into the path loss estimation to provide a realistic range rather than a single-point value.

## 7.6. Path loss calculation

- **Without tolerance:**
  **Path Loss (dB)** = Tx Power (dBm) - RSSI (dBm)
  **Example**: Tx Power = 0 dBm, RSSI = -75 dBm → Path Loss = 75 dB

- **With tolerance:**
  **Path Loss (dB) =** Tx Power (dBm) - RSSI (dBm) ± Tx Power Tolerance (dB)
  **Example:** Tx Power = 0 dBm, RSSI = -75 dBm, Tx Power Tolerance = ±2 dB → Path Loss = 75 ± 2 dB

### 7.6.1. Sampling duration

- Each placement per distance takes roughly 45-60 seconds (in a controlled environment).
- **Note:** The app scans in foreground only.
- Share results via AirDrop or email.

## 7.7. Controlled RF chamber (recommended)

- Provides a clean, repeatable, line-of-sight (LoS) setup
- Recommended for baseline calibration

## 7.8. Semicontrolled indoor (office setup)

- Use a low-reflection environment.
- Use tripods or non-metallic stands.

2026-09-01 | Copyright © 2026 Apple Inc. All Rights Reserved.

- Mark distances at 20 cm, 50 cm, and 1 m.

## 7.9. Edge cases and limitations

- RSSI values fluctuate; average over N samples to improve accuracy.
- iPhone and iPad antenna performance varies by model; account for variance in measurements.

## 7.10. Security and privacy

- **Collects only engineering metadata.**
- **Captures no PII.**
- **Complies with App Transport Security for uploads.**

## 7.11. Environmental factors

- **Multipath propagation:** Reflected signals can cause RSSI fluctuation.
- **Obstructions:** Human bodies, walls, and other objects attenuate or reflect signals.
- **Antenna orientation:** Misalignment between devices alters signal strength.
- **Distance:** Path loss increases non-linearly with distance.
- **Interference:** 2.4 GHz noise from Wi-Fi or other sources may degrade readings.
- **Elevation mismatch:** Signal strength varies if devices aren't level.

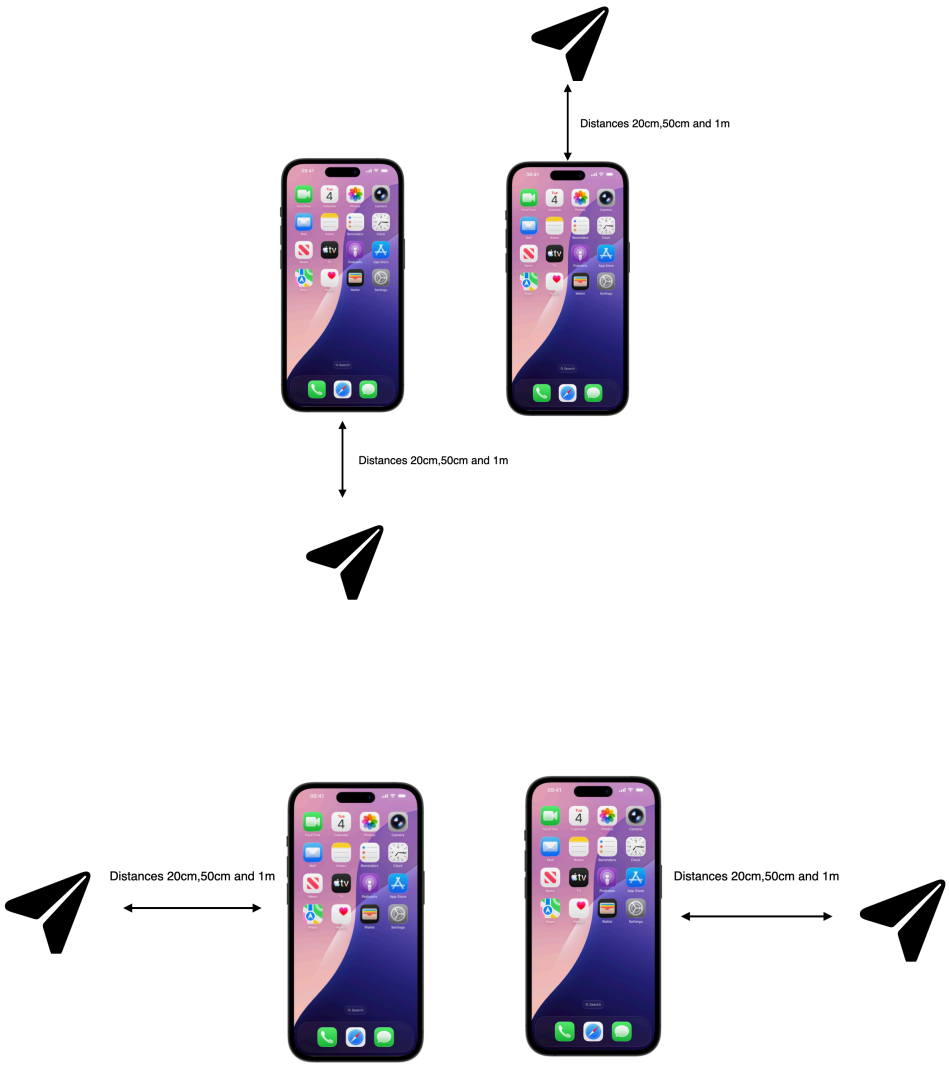
## 7.12. Device-specific factors

- **Model and antenna design:** Different iPhone and iPad models may yield different RSSI behavior.
- **Phone case or material:** Some enclosures attenuate BLE signals.
- **Accessory Tx Power calibration:** Uncalibrated transmit power skews results.
- **PHY used:** BLE 1M, 2M, and Coded PHYs behave differently.

## 7.13. Measurement strategy

- **Sample count:** Too few samples increase variance.
- **Sampling rate:** Too fast during movement adds instability.
- **App state:** Background can delay capture.
- **Device movement:** Changing positions between samples invalidates conclusions.

2026-09-01 | Copyright © 2026 Apple Inc. All Rights Reserved.




## 7.14.Interpretation pitfalls

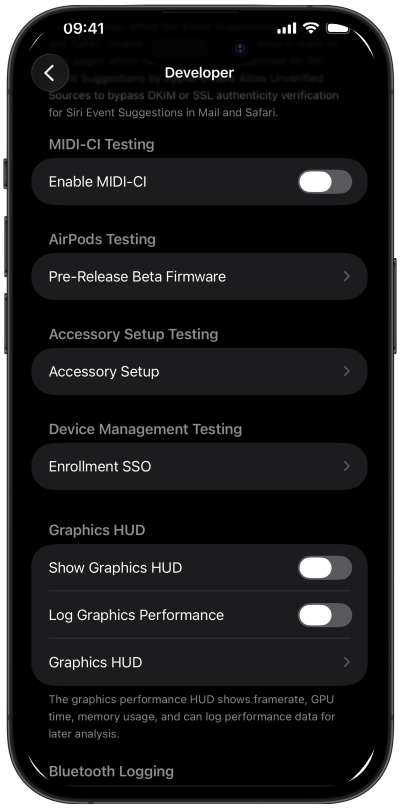
- **Assuming linear distance-RSSI relationship:** BLE signal propagation is logarithmic.
- **Ignoring Tx Power tolerance:** May result in overconfidence in precision.
- **Mixing environments:** Results are not comparable across test sites.
- **Overlooking PHY settings or connection state:** Affects signal-strength reporting.

2026-09-01 | Copyright © 2026 Apple Inc. All Rights Reserved.

## 7.15. Test setup instructions and data collection

- A square or round table with 1 m of clearance is preferable, but a rectangular table is acceptable so long as there is 1 m of clearance on at least one side. Mark the center point and the 20 cm, 50 cm, and 1 m points.
- The table should be at least 1 m away from all walls and reflective surfaces.
- The bottom of the table should have no metal or reflective surface.
- If you don't have a flat surface to fit or hold the accessory, you can rotate the accessory to achieve the correct orientation, so long as you can maintain the relative position of the phone and accessory.
- Follow the same placement as shown in Section 7.13, and repeat the test for all the *required distances 20, 50, and 100 cm respectively*.
- Run the test across multiple models of each supported device (iPhone and/or iPad) and repeat testing to confirm stable results. If RSSI offsets vary, submit the log file with the lowest value through the Portal to help ensure consistent performance across all platforms.

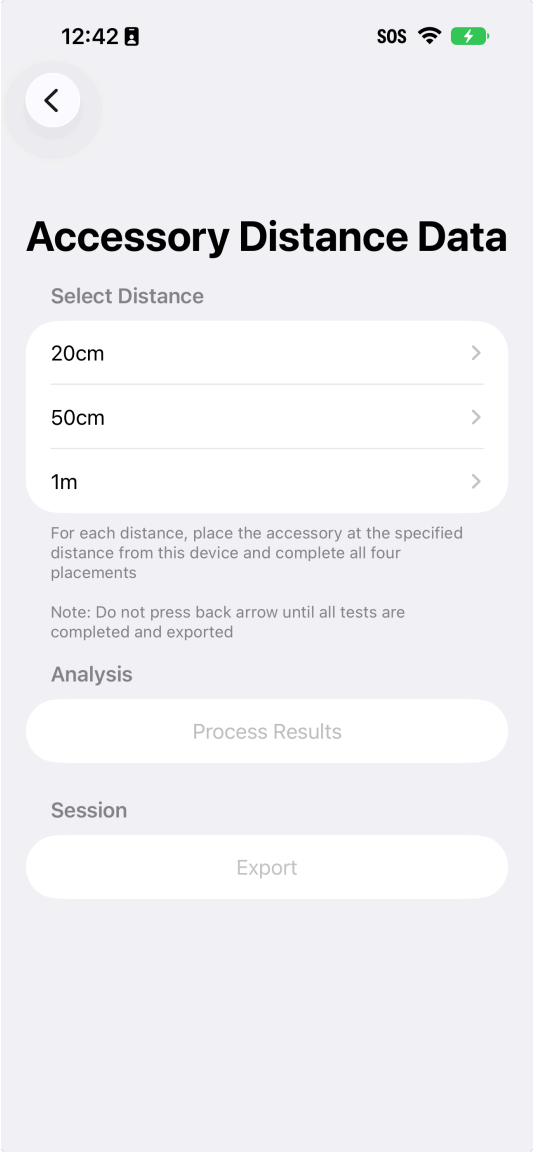
## 7.16. Enabling the data-collection tool

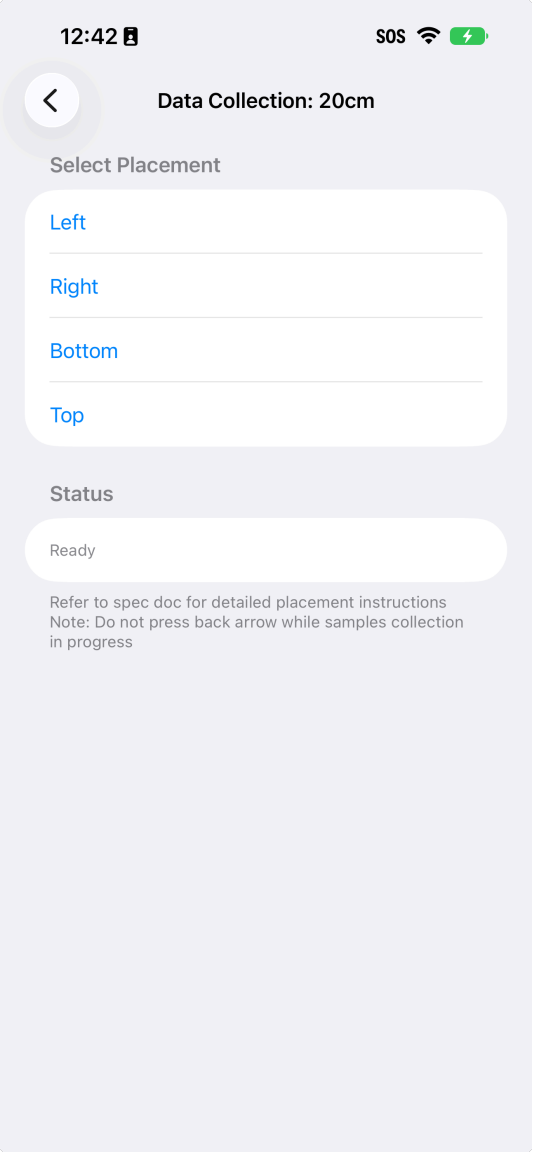



You can enable the self-certification data-collection tool using the provided developer profile on the beta

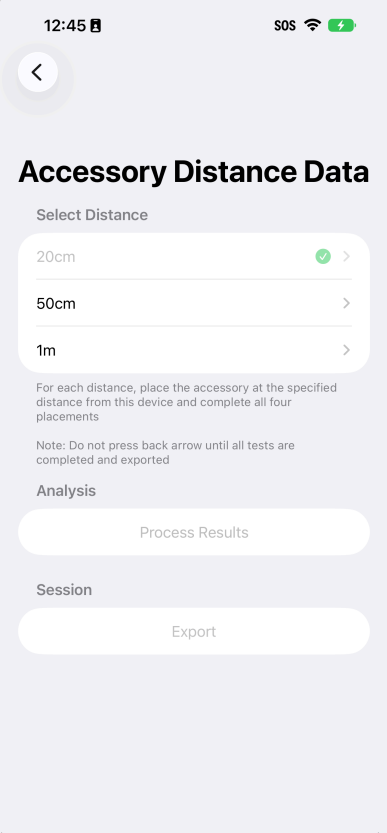
2026-09-01 | Copyright © 2026 Apple Inc. All Rights Reserved.

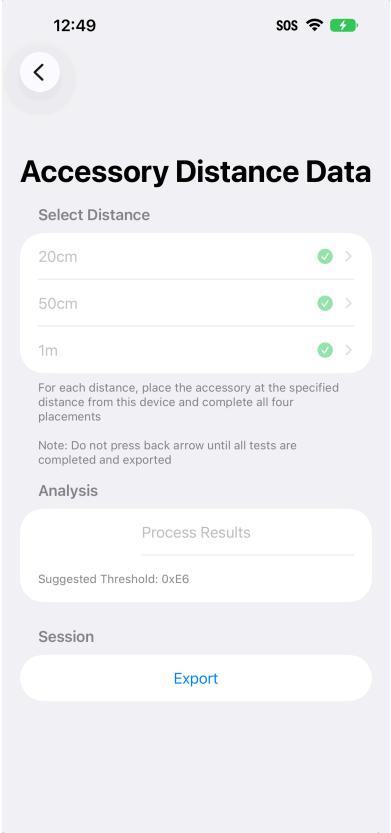
release of iOS 26.5. Once you install the profile and reboot the phone, the Accessory Setup setting is available in the Developer settings, as shown in the following screenshots.

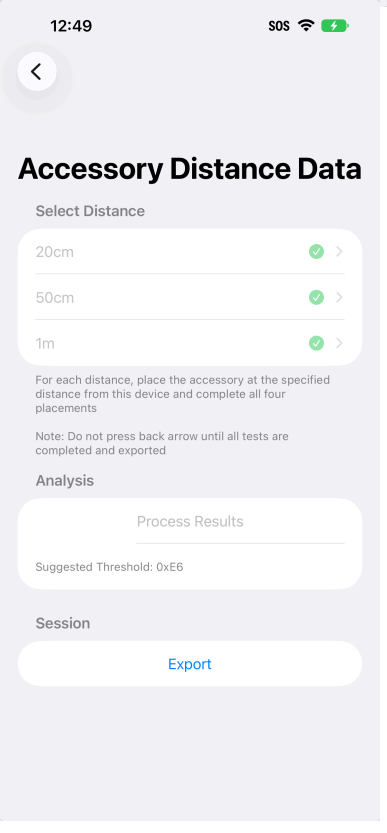


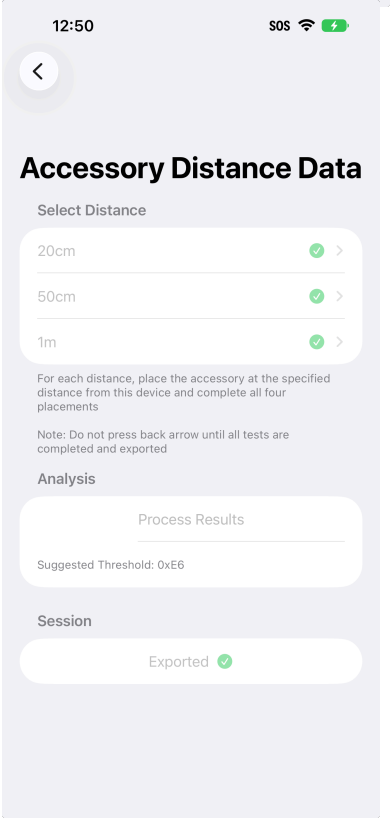




2026-09-01 | Copyright © 2026 Apple Inc. All Rights Reserved.



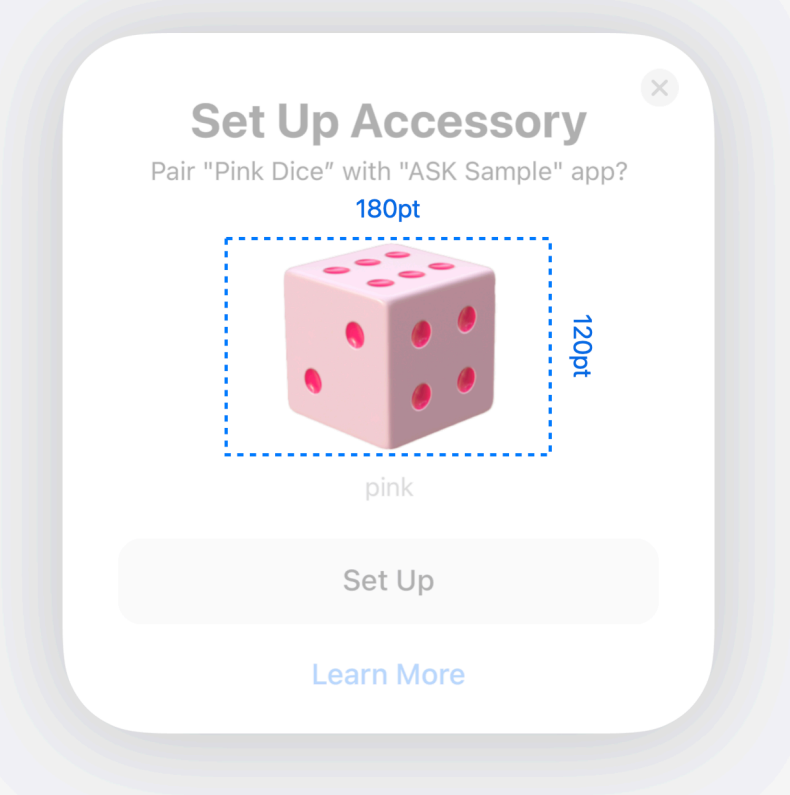

2026-09-01 | Copyright © 2026 Apple Inc. All Rights Reserved.

# 8. Asset details

Full asset specifications and reference materials are available in the official WWDC24 session for AccessorySetupKit: https://developer.apple.com/videos/play/wwdc2024/10203/

Format
Background: Transparent
Colorspace: sRGB
Size
- Height: 120 pt (@3x)
- Width: 180 pt (@3x)

Padding
- Square or evenly proportioned accessories
  - -Top/Bottom margins: 15% of overall area
  - -Left/Right margins: 15% of overall area
- Long accessories
  - -Top/Bottom margins: 20% of overall area
  - -Left/Right margins: 5% of overall area
- Tall accessories
  - -Top/Bottom margins: 5% of overall area
  - -Left/Right margins: 20% of overall area




2026-09-01 | Copyright © 2026 Apple Inc. All Rights Reserved.

# 9. Revision history

| Version | Date | Notes |
|---|---|---|
| R1 | 2026-05-28 | Release R1 |
| R1.1 | 2026-09-01 | Release R1.1, Added support for accessory self-certification and registration on iPadOS. |

2026-09-01 | Copyright © 2026 Apple Inc. All Rights Reserved.

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, the Apple Logo are trademarks of Apple Inc., registered in the U.S. and other countries. iOS is a trademark or registered trademark of Cisco in the U.S. and other countries and is used under license.

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.