

Accessory Proximity-Triggered Pairing Specification

Release R1



Accessory Proximity-Triggered Pairing Specification	1
Release R1	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. Sub Type	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

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.Semi-Controlled 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

7.16.Enabling Data collection tool	29
8. Asset details	32
9. Revision History	33

1. Introduction

The document 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.

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.

1.1. Terminology

Throughout this document, these terms have specific meanings:

- The term *Apple device* is used to refer to an iPhone (running iOS).
- The term *accessory* is used to refer to any product intended to interface with a device through the means described in this specification.
- 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 is used to refer to the Bluetooth Special Interest Group (SIG) which is a network of member organizations that are the caretakers and innovators of Bluetooth® technology.

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.

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 cannot support BLE advertising then proximity setup is not possible.

2.3. Operation

When an accessory is in pairing or setup mode, it advertises via Bluetooth Low Energy (BLE) beacon with a specific payload to indicate it's in this mode. This beacon can be picked up by a nearby Apple device and if close enough to the accessory, the Apple device will prompt 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 for the user.

2.4.2. Apple Server

Information about the accessory is stored on Apple servers; this could include the product type, the product image, associated app bundle and the supported services that are provided 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 assigned by the Bluetooth Special Interest Group (SIG) to a specific company or organization. This identifier is used in the Manufacturer Specific Data element within BLE advertising packets.

Each company or organization is typically assigned a unique Company ID to avoid conflicts and ensure interoperability between different Bluetooth devices. The purpose of having unique identifiers is to allow devices to recognize and interpret the Manufacturer Specific Data correctly.

2.4.4. Accessory ModelID

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 models within a product family, such as by color, size, or other variation.

2.4.5. RSSI Offset

For this release, threshold that triggers the pairing card to appear is set at 20 cm, which is consistent with all other Apple accessories. This is calculated by the self certification tool.

Example: If the RSSI on the phone for the accessory is -55dBm, use that as the last byte of the payload. The RSSI Offset byte is the two's complement representation of the RSSI value encoded as an unsigned byte. For example, if the RSSI is calculated as -55 dBm then use $256 - 55 = 201$, which is 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 advertisement packets for discovery. The format of these packets is authoritatively defined in the Bluetooth specifications. To simplify implementations, this section provides a summary of the specific BLE packet formats used by this document.

Bluetooth Low Energy (BLE) advertisements are delivered via Bluetooth data link layer packets 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 Sub-Types
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

2.6. Supported Modes

At registration, provide a 1-byte supported mode value. This is used to determine the capabilities of the accessory and provide the appropriate user experience flow. The details for the supported modes are listed below:

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

Accessory supports a custom service that requires the Manufacturer's application to complete the setup and to enable the functionality of the accessory. This could also be used if the accessory is advertising a standard Bluetooth service defined by the Bluetooth SIG. Use this mode only when the accessory exclusively supports Bluetooth LE for connectivity and does not 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)

Accessory supports a service that is specified by Apple and is built into the system. Some examples of a system service include the Human Interface Device (HID) and Apple notification center service (ANCS). After completing the Bluetooth pairing and authentication, the user will be guided through the setup for the services supported by the accessory. If there is a companion app that is required, the user will be prompted 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 + 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. If this is enabled iOS will connect directly over Bluetooth classic and setup supported services.

Table below provides details of the supported services byte that the accessory needs to advertise to provide the best experience for the setup based on the supported services.

Table 2-2 Supported Modes

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

2.7. Sub Type

This document defines the following sub-types. Sub-types not defined by this document are reserved for definition by other documents.

Table 2-3 — Sub-Type 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.

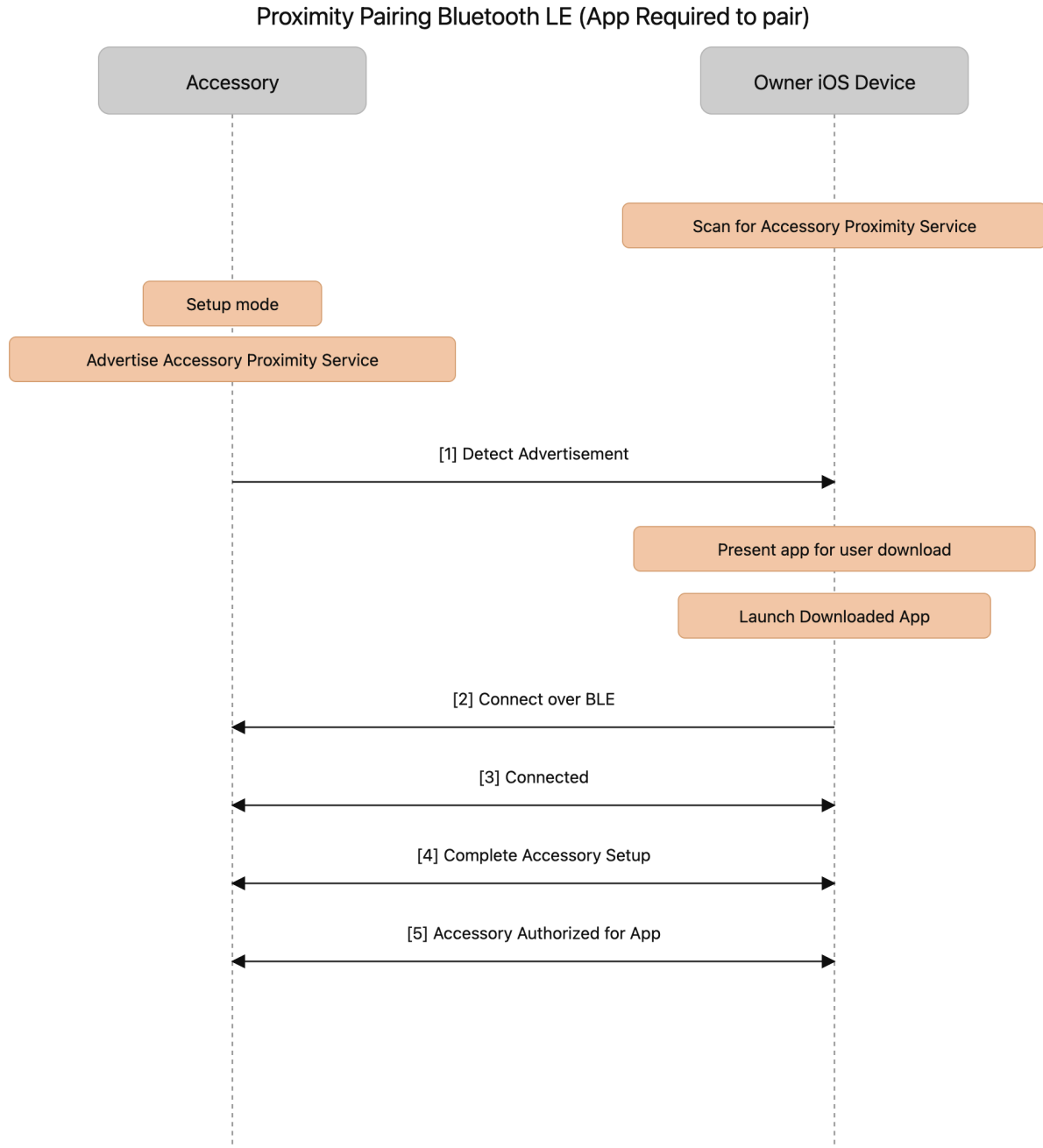
2.8. Accessory Name

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 will be used as the title of the proximity card. Details on the friendly name are provided in section 5.

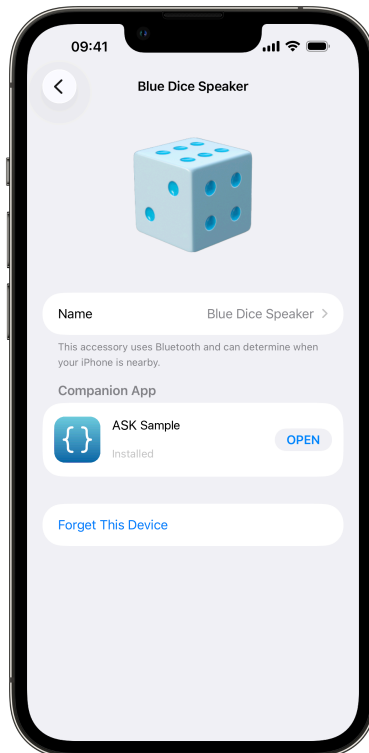
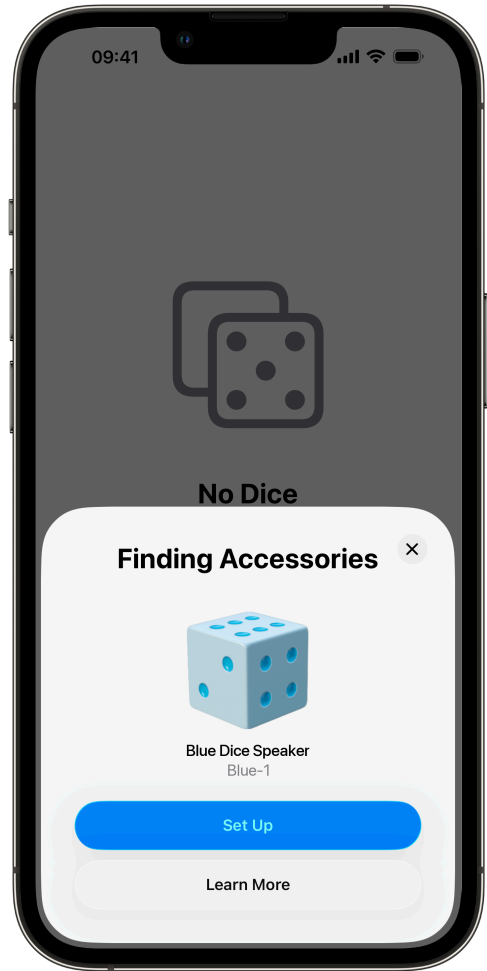
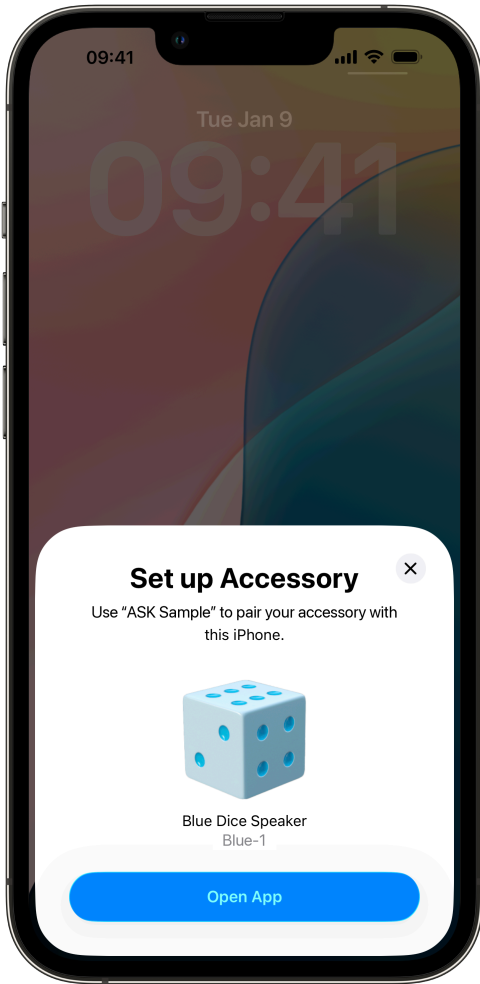
2.9. Pairing Flow

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

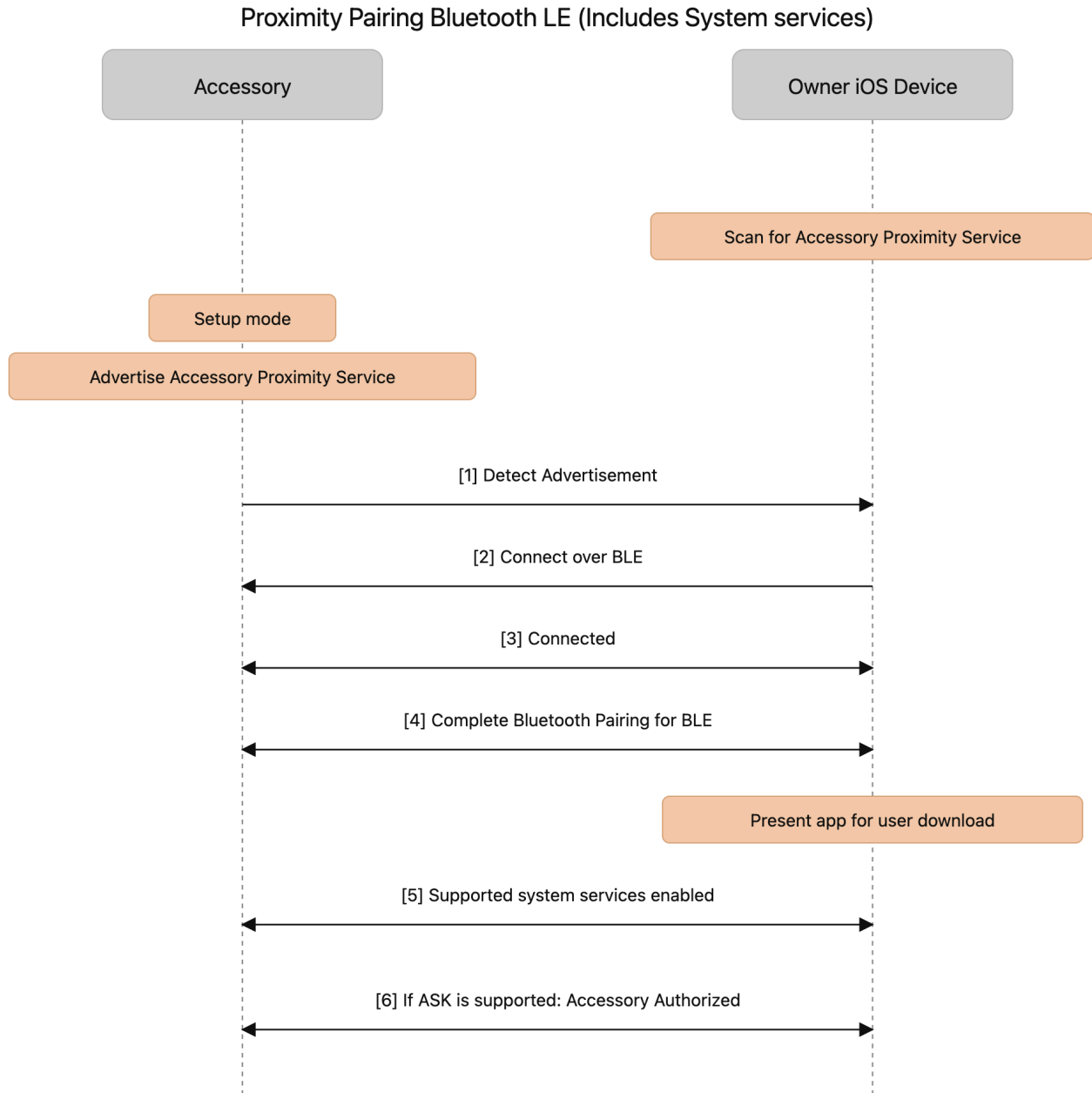
2.9.1. Bluetooth LE (App Required to pair)



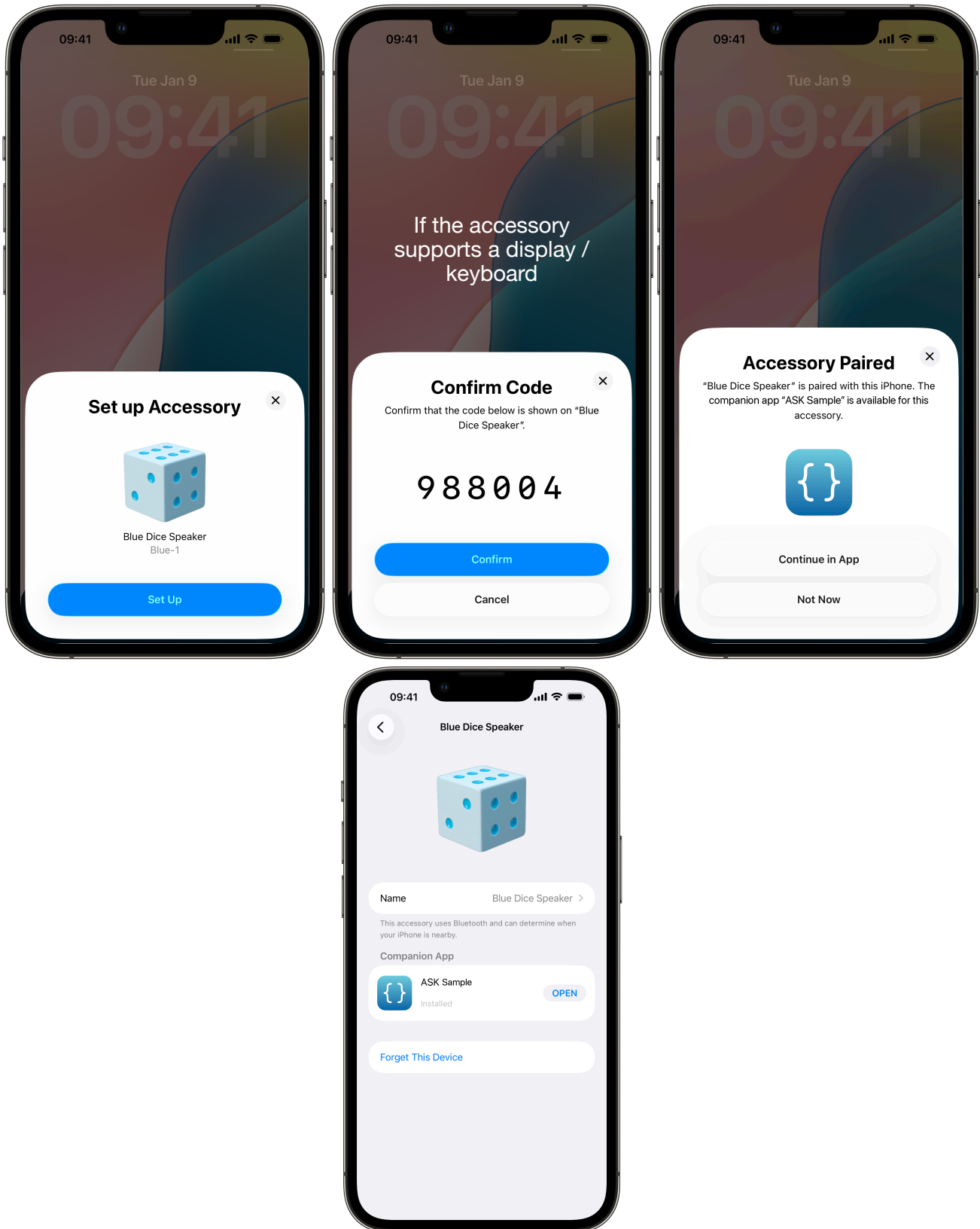
When the accessory does not support Bluetooth pairing and the setup is controlled by the application, it is required to follow this flow for the most optimal user experience.



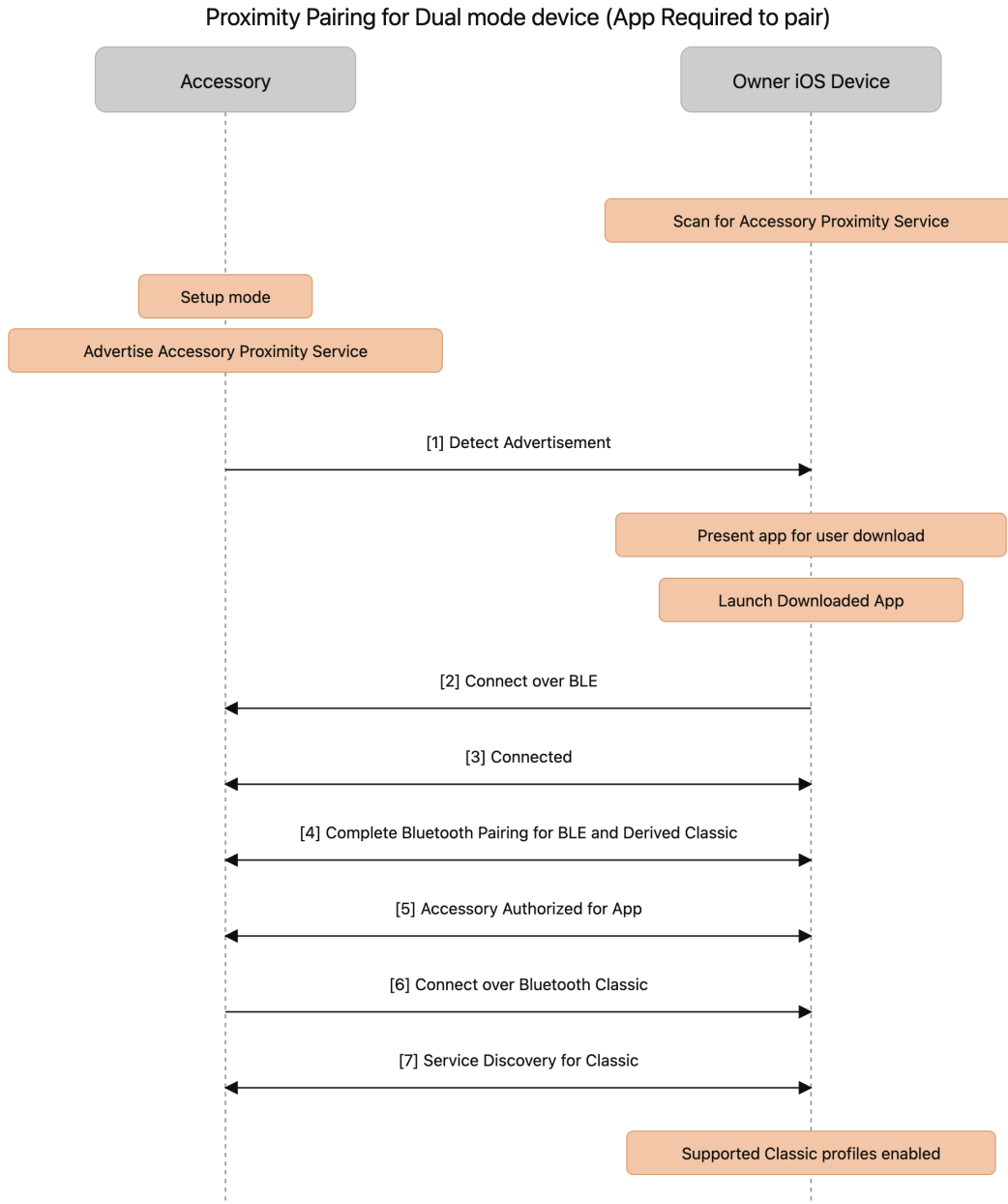
2.9.2. Bluetooth LE (Includes System services)



When the accessory supports Bluetooth pairing it is required to follow this flow for the most optimal user experience.

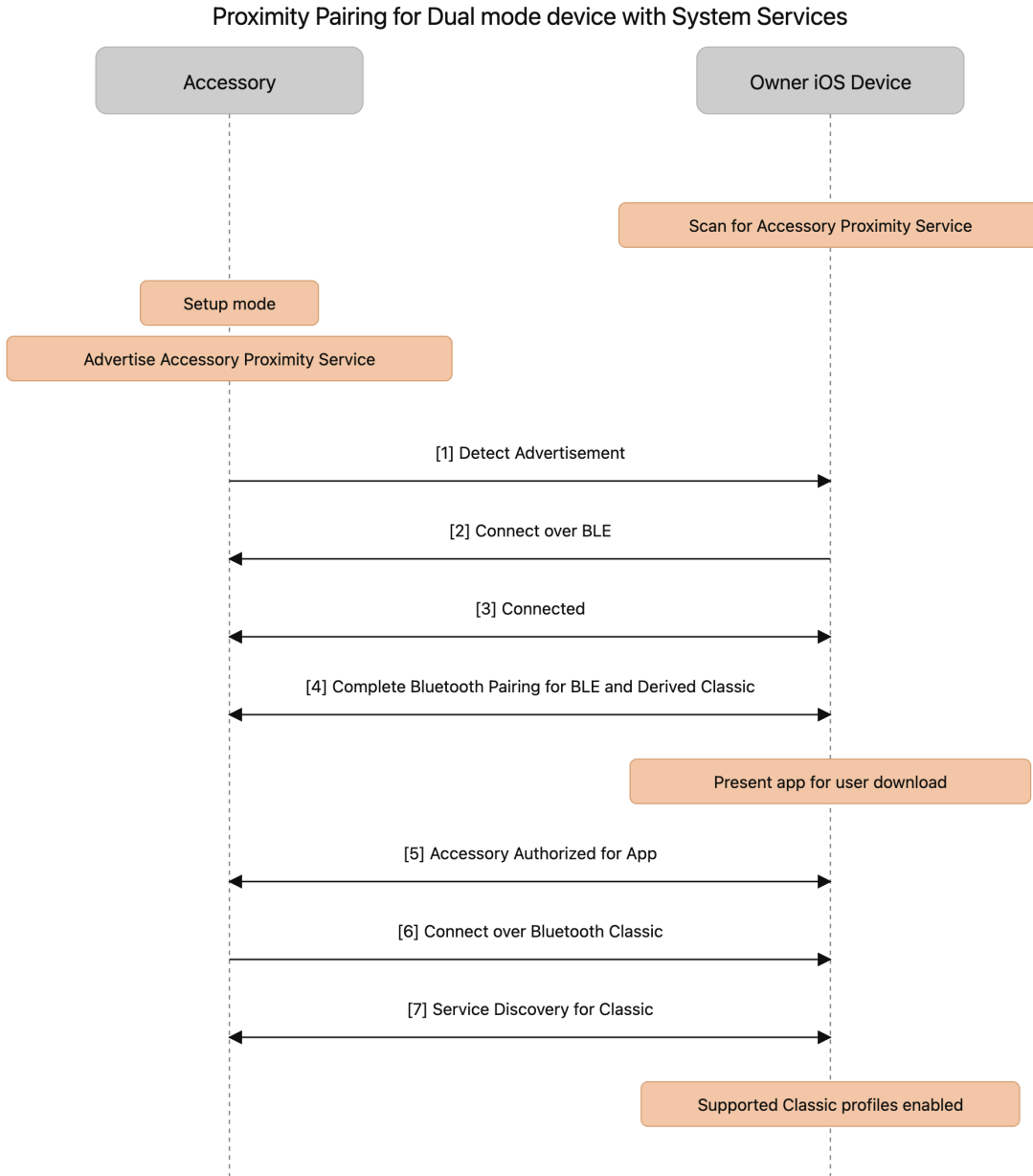


2.9.3. Dual mode device (App Required to pair)



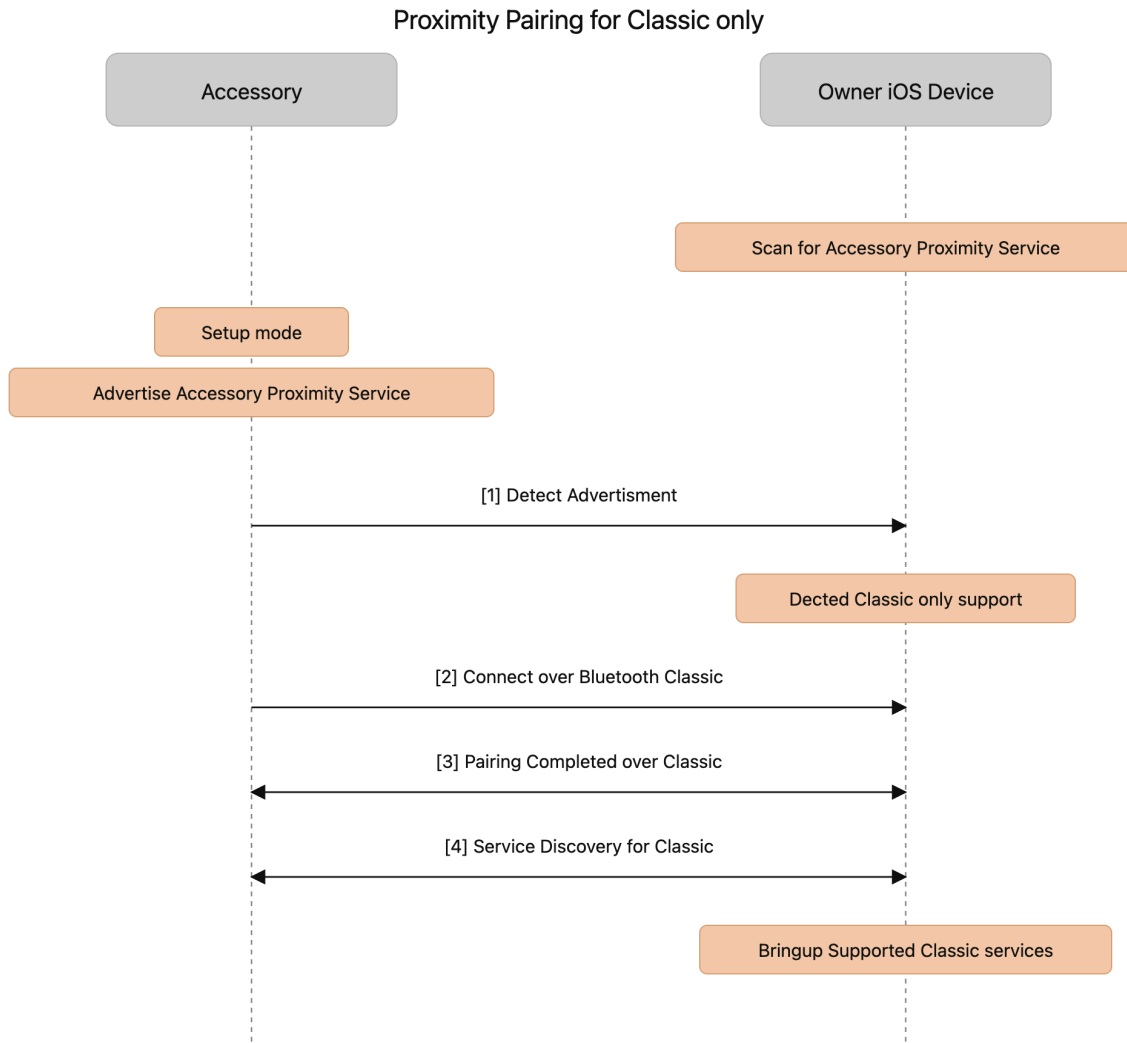
This mode follows the same flow as BLE-App required, if the accessory supports GATT over classic then Bluetooth pairing is required. Use this flow only if there are custom modes needed in setup that cannot be achieved without specific instructions.

2.9.4. Dual mode device (Includes System services)



When the accessory supports Bluetooth pairing it is required to follow this flow for the most optimal user experience.

2.9.5. Accessory Supports Classic Only



When the accessory does not support BLE connections and only supports Bluetooth classic pairing it is required to follow this flow for the most optimal user experience. No BLE functionality will be enabled and connections will not be possible.

3. Requirements

3.1. Overview

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

3.2. General

- If an accessory is powered on and is not 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 a timeout or on successful pairing
- If an accessory is reset by the user, it should enter setup mode automatically.

4. Bluetooth Requirements

4.1. Overview

Bluetooth Low Energy (LE) is used 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 $\geq +4\text{dBm}$. 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 100ms. 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.

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.

5. Product registration and self-certification

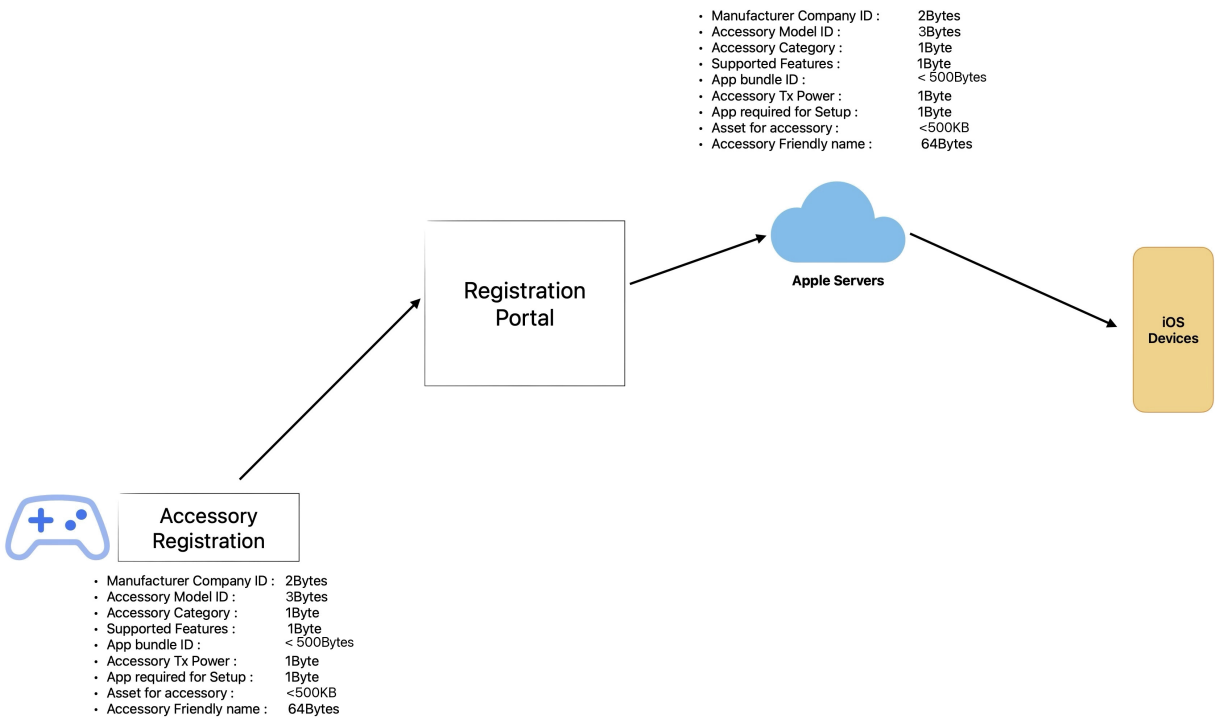
5.1. Requirements

To enable proximity pairing in an accessory, developers must submit the information shown in the table below and complete self-certification through the Portal. Once self-certification is complete, this information will be synced to Apple’s servers and the accessory will be able to 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

A developer profile is provided to allow developers to 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](#).



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.

For this release we have provided a test company ID that we would require your accessory 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 of your application, please use the following key for the system to match the payload. Ensure that the Info.plist has NSAccessorySetupSupports and Bluetooth so it adopts AccessorySetupKit (<https://developer.apple.com/documentation/accessorysetupkit?language=objc>)

```
<key>NSAccessorySetupProximityServices</key>  
<array>  
</array>
```

Inside the array the following fields can be provided :

ModelID, CompanyID, Image, BundleID, AccessoryName

ModelID: Any 24-bit ID determined by the manufacturer. 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.

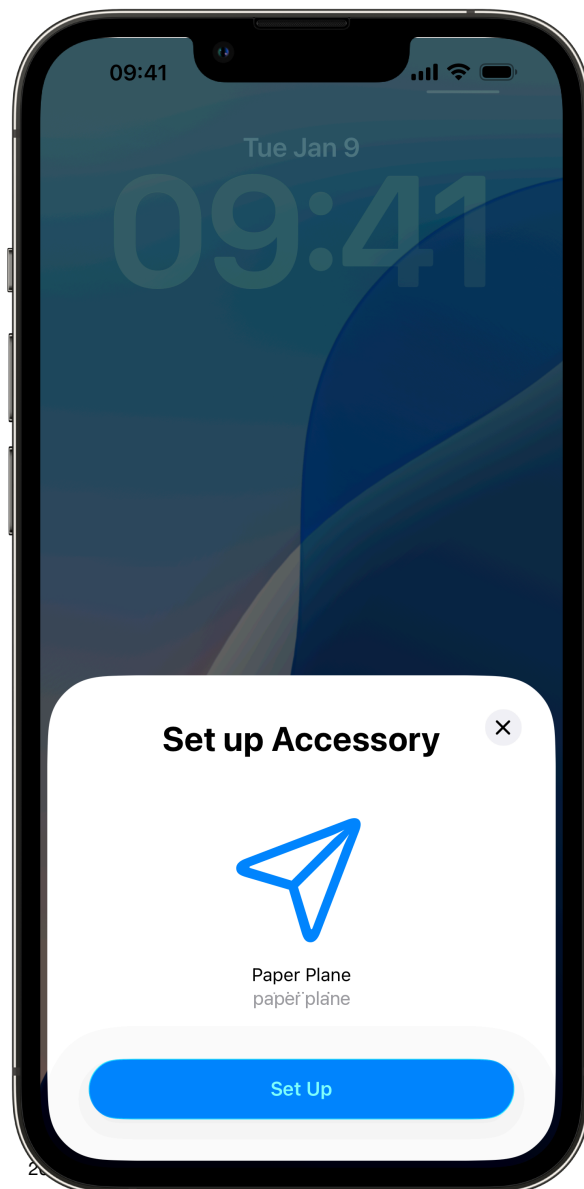
BundleID: The bundle ID of the application that will be authorized by AccessorySetupKit.

Image: An SF Symbol that best matches your accessory, displayed on the pairing card when the accessory is discovered.

AccessoryName: A friendly name displayed as the title of the pairing card.

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>
```



</array>

Pairing Card Display:

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

Yes, instantiate an `ASAccessorySession` and 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 received 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

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 CoreBluetooth APIs in the iOS development environment. The specification is intended for use in 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.
- 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. CoreBluetooth exposes limited RF parameters; however, path loss can be estimated using the received signal strength indicator (RSSI) and transmitted power (Tx Power), if available.

CoreBluetooth 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

- BLE accessory is advertising.
- Accessory includes Tx Power in its advertisement packet.
- RSSI can be sampled during scanning.
- iOS app is running in the **foreground**.
- **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 should be integrated into the computation model or reported alongside the result for transparency.

- Advertisement interval should be < 100 ms

7.5. Data Collection

- Install the provided Accessory Proximity-Triggered Pairing Test Profile.
- App uses `CBCentralManager` to scan peripherals.
- Accessory will be advertising 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), it should be factored 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 \rightarrow Path Loss = 75 dB
- **With tolerance:**
Path Loss (dB) = Tx Power (dBm) - RSSI (dBm) \pm Tx Power Tolerance (dB)
Example: Tx Power = 0 dBm, RSSI = -75 dBm, Tx Power Tolerance = ± 2 dB \rightarrow Path Loss = 75 ± 2 dB

7.6.1. Sampling Duration

- Each placement per distance roughly takes 45-60 seconds (in a controlled environment)
- **Note:** App is designed to scan 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. Semi-Controlled Indoor (Office Setup)

- Use a low-reflection environment.

- Use tripods or non-metallic stands.
- 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 antenna performance varies by model and should be accounted for in measurements.

7.10. Security and Privacy

- **Only engineering metadata is collected.**
- **No PII is captured.**
- **Complies with App Transport Security for uploads.**

7.11. Environmental Factors

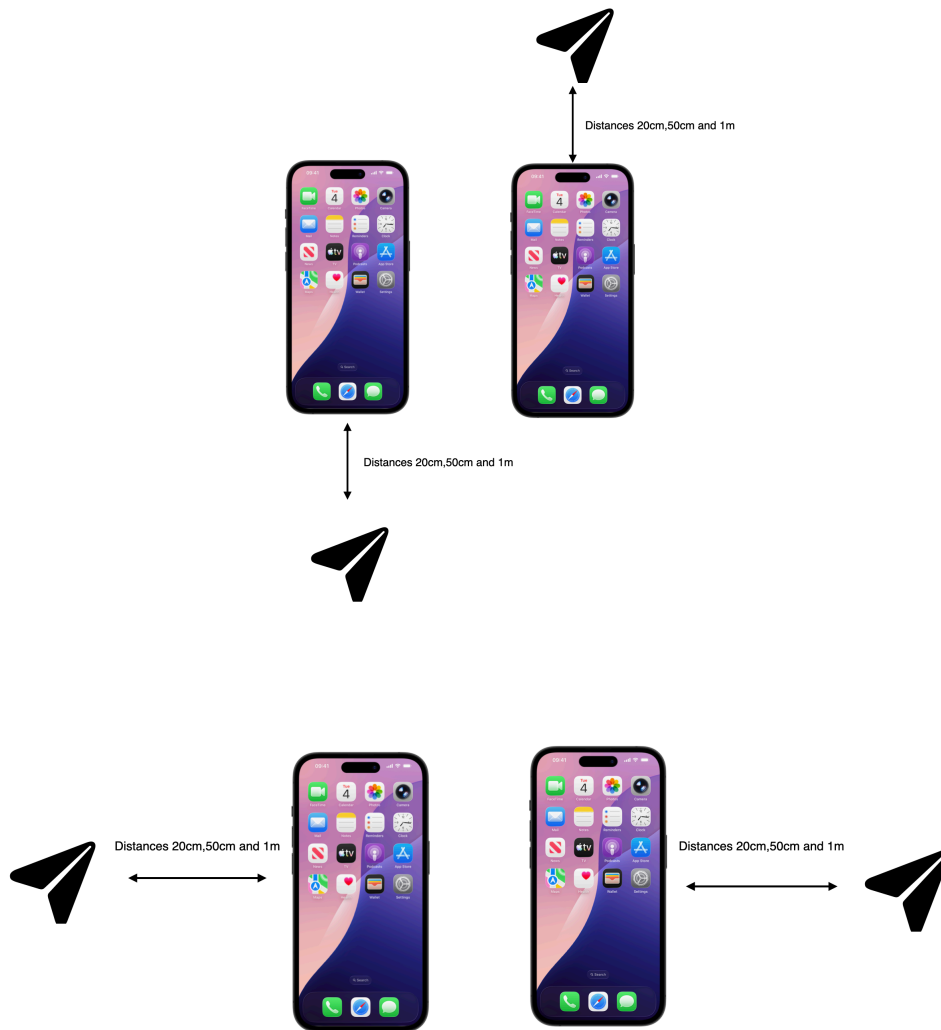
- **Multipath propagation:** Reflected signals can cause RSSI fluctuation.
- **Obstructions:** Human bodies, walls, and other objects attenuate or reflect signals.
- **Antenna orientation:** Misalignment between devices alters signal strength.
- **Distance:** Path loss increases non-linearly with distance.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

7.12. Device-Specific Factors

- **Phone model and antenna design:** Different iPhones 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.



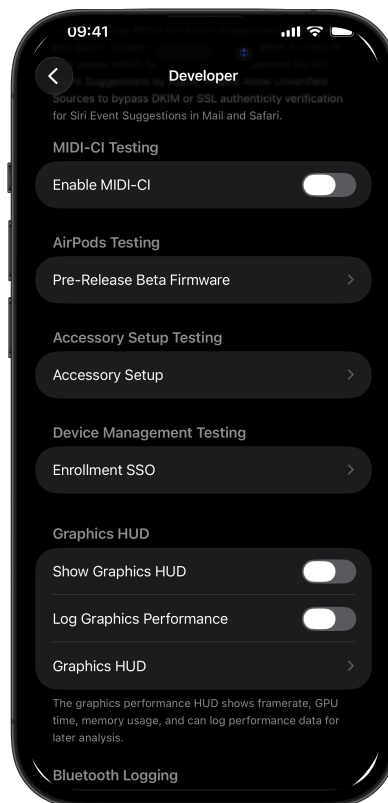
7.14. Interpretation Pitfalls

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

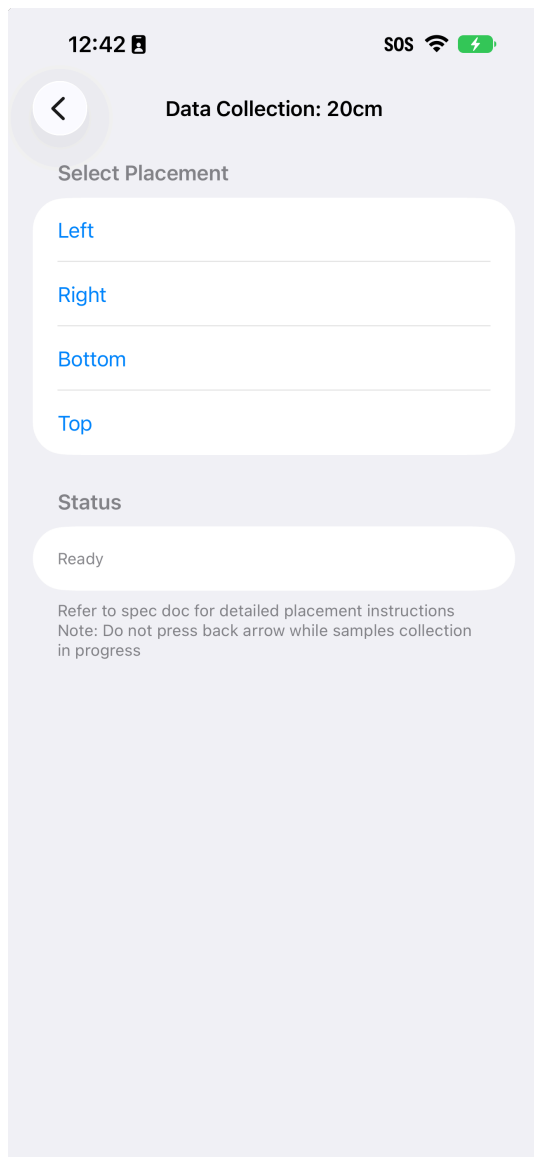
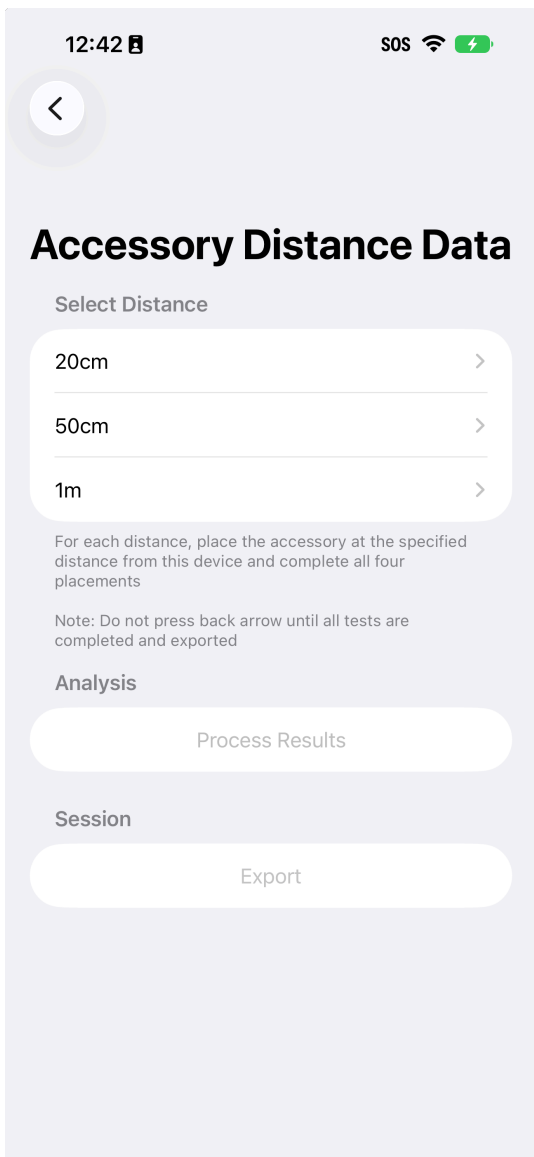
7.15. Test Setup Instructions and Data Collection

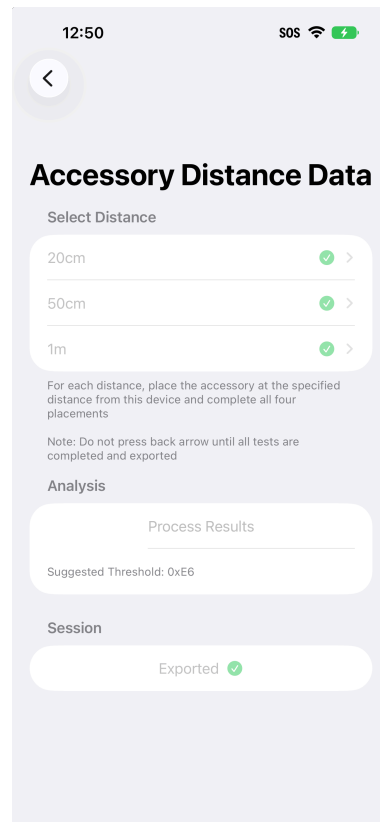
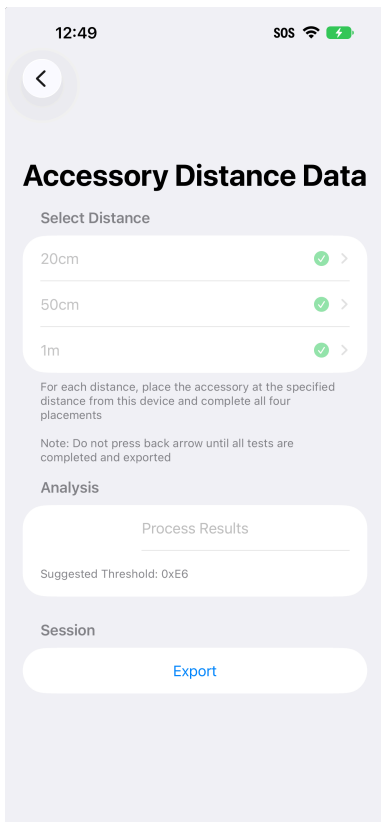
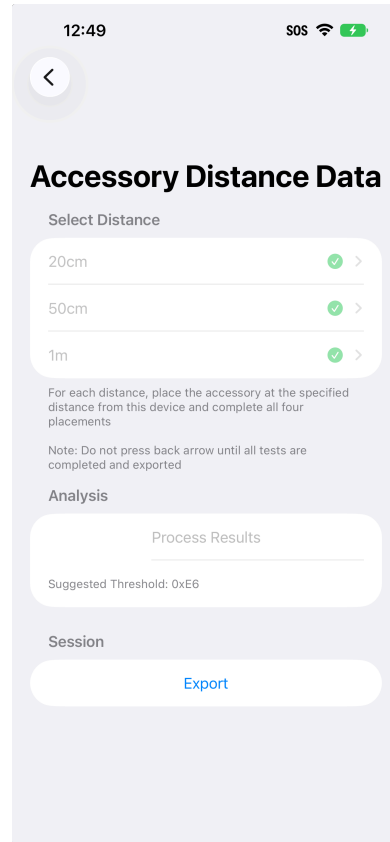
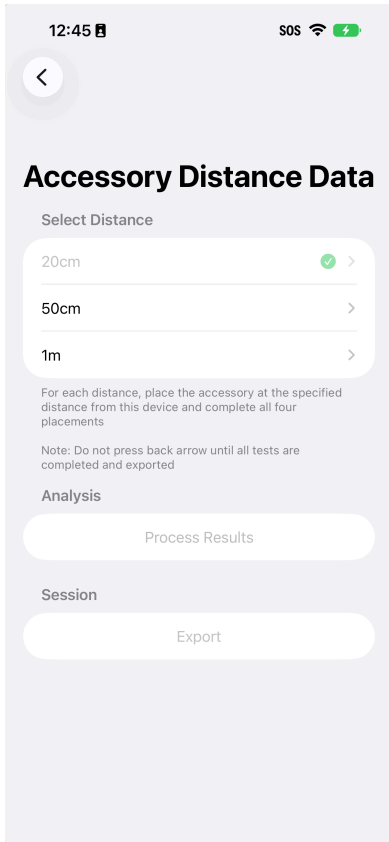
- A square or round table with 1m clearance is preferred, but a rectangular table is acceptable so long as there is 1m clearance on at least one side. Mark the center and 20cm,50cm,1m points.
- Table should be at least 1m away from all walls and reflective surfaces.
- Bottom of the table should have no metal or reflective surface.
- In case where the accessory does not have a flat surface to fit on or be held on, it is acceptable to rotate the accessory in order to achieve the correct orientation, so long as relative position of phone and accessory can be maintained.
- Follow the same placement as shown in Section 7.13 and repeat the test for all the desired distances 20,50 and 100 cm respectively.

7.16. Enabling Data collection tool



The self-certification data collection tool can be enabled using the provided developer profile on the beta release of iOS 26.5. Once the profile is installed and the phone is rebooted the "Accessory Setup" setting is available in the Developer Settings. (See screenshot above)





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

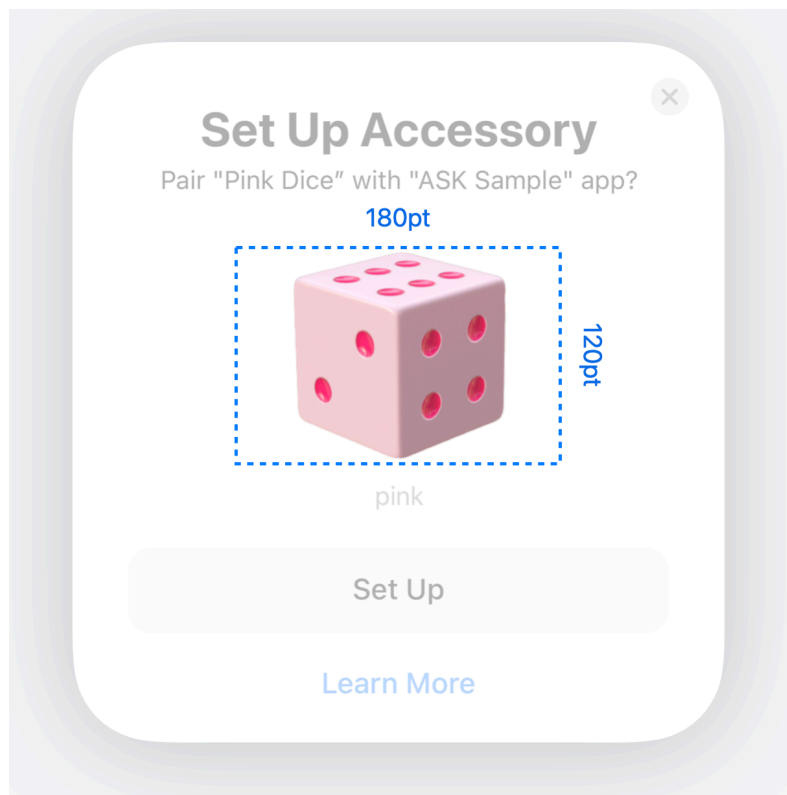
-Top/Bottom margins: 20% of overall area

-Left/Right margins: 5% of overall area

Tall accessories

-Top/Bottom margins: 5% of overall area

-Left/Right margins: 20% of overall area



9. Revision History

Version	Date	Notes
R1	2026-05-28	Release R1
R1	2026-06-11	Added link for downloading the profile



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.