

Bluetooth Accessory Design Guidelines for Apple Products

Release R6

Contents

Introduction 5

Organization of This Document 5

General Requirements 6

Conformity With Bluetooth Specifications 6

Enhanced Data Rate 6

Sniff Mode for Low Power Consumption 6

Role and Topology Management 7

Extended Inquiry Response 7

Secure Simple Pairing 8

Profiles 8

Device ID Profile (DID) 8

Hands-Free Profile (HFP) 9

Message Access Profile (MAP) 11

Audio/Video Remote Control Profile (AVRCP) 11

Advanced Audio Distribution Profile (A2DP) 14

Audio Routing 15

Audio Data Received via HFP Profile 16

Audio Data Received via A2DP Profile 16

Bluetooth Low Energy 17

Role 17

Advertising Channels 17

Advertising PDU 17

Advertising Data 17

Advertising Interval 18

Connection Parameters 18

Privacy 19

Permissions 19

Pairing 19

Services 19

GATT Server 20

Apple Protocols and Developer Support 21

Apple Bluetooth Development Mailing List 21

The iPod Accessory Protocol 21

Headset Battery Level Indication 21

 HFP Command AT+XAPL 22

 HFP Command AT+IPHONEACCEV 22

Document Revision History 24

Tables

General Requirements 6

Table 1-1	SubBand Codec Information Elements for Apple products	14
Table 1-2	MPEG-2/4 AAC Codec Information Elements for iOS devices	14
Table 1-3	AAC audio packet for iOS devices	15

Introduction

This document presents design guidelines for hardware accessories that use Bluetooth transport to communicate with Apple products, including Mac, iPhone, iPad, and iPod touch models.

Note: This document is a **draft** and is subject to change without notice.

To be compatible with Apple products, both current and future, Bluetooth accessories should follow the specifications in this document.

IMPORTANT: This document uses the term “Apple product” to refer generically both to Mac (Apple computers that run OS X) and to iPod, iPhone, and iPad models. Among the latter products, those that run iOS (Apple’s mobile operating system) are also referred to as “iOS devices.” Specifications in this document that are designated for iOS devices apply only to those products.

Organization of This Document

The specifications in this document are presented in three chapters:

- “[General Requirements](#)” (page 6) relates the design of hardware accessories that use Bluetooth transport to the general Bluetooth specification.
- “[Apple Protocols and Developer Support](#)” (page 21) references an Apple protocol that can extend accessory capabilities beyond those supported by standard Bluetooth profiles.

General Requirements

Conformity With Bluetooth Specifications

Every Bluetooth accessory that is compatible with an Apple product should support the Bluetooth Core Specification Version 2.1 + EDR or higher. This specification introduced the important security feature Secure Simple Pairing as well as Extended Inquiry Response.

Enhanced Data Rate

The Enhanced Data Rate (EDR) feature introduced in the Bluetooth 2.0 specification enables Bluetooth accessories to communicate more efficiently. To provide backward capability, the Bluetooth SIG made EDR an optional feature. Apple strongly recommends the use of EDR as much as possible for the following reasons:

- It provides higher data rates compared to Basic Data Rate (BDR).
- It communicates more efficiently, transferring more data bits per unit of time.
- It reduces the power consumption used per bit transferred.
- It coexists better with WiFi using a shared antenna because it frees up more air time for WiFi.
- It improves performance in multipoint configurations.

Sniff Mode for Low Power Consumption

Minimizing power consumption is critical for all mobile devices. Therefore, every Bluetooth accessory that is compatible with an Apple product:

- Should support and should request Bluetooth Sniff Mode
- Should accept requests for Sniff Mode and support all valid parameters listed in the Bluetooth specification.

Bluetooth accessories that are compatible with Apple products should also use Sniff Mode as much as possible, especially when there is little or no data being transmitted over the Bluetooth link. Besides its power consumption advantages, Sniff Mode enables better antenna sharing with WiFi.

The Sniff Mode parameters are specific to the usage model and Bluetooth Profile. The Apple product expects the accessory to request Sniff Mode with appropriate parameters for a specific usage. If the accessory does not send such a request, the Apple product may send a Sniff Mode request. When the Apple product sends a request for Sniff Mode, the remote device should accept the request and its parameters without negotiation.

If it sets the Sniff Mode parameters, the Bluetooth accessory should set the sniff Interval to less than a third of the Bluetooth baseband Link Supervision Timeout. This makes the Bluetooth link less susceptible to interference. To improve link robustness, the accessory should use a shorter sniff Interval instead of multiple sniff attempts. Links with a sniff Interval of 1 second or more make the slave device open up a large correlation window, which has to be taken into account when calculating the number of sniff attempts. With sniff intervals shorter than 1 second, multiple sniff attempts can improve link robustness but will increase power consumption.

Role and Topology Management

Every Bluetooth accessory that is compatible with an Apple product should:

- Accept a request for Role Switch from an Apple product.
- Continue with the connection when the Apple product rejects a request for Role Switch.

In a Bluetooth connection, one device is the master and the other the slave. The master can have multiple slaves, thus forming a piconet. The master device can also be a slave role to another master, creating a scatternet. Such a scenario creates complications since the device has to alternate between the two piconets and thus wastes valuable bandwidth. Managing the topology of the network is therefore important for maximum performance. The Apple product can request a Role Switch, depending on its current topology, and the remote device should accept the request. The Apple product may also reject a request for a Role Switch because of topology concerns. Having a suboptimal topology may degrade the audio quality and the user's experience.

Only when it is maintaining multiple links, either Bluetooth or WiFi, will the Apple product request or deny role switches. Hence, it will grant a role switch if there is no reason for the Apple product to be master. It is expected that the accessory will behave the same, only trying to be master when there is a legitimate reason. The accessory should not always request to be master by default if there is no need in the system topology to do so. If later the accessory needs to be master in order to maintain multiple links, it should ask to be master at that time.

Extended Inquiry Response

Every Bluetooth accessory that is compatible with an Apple product should provide the following information in its Extended Inquiry Response packet:

- The Local Name of the product (Complete or Shortened).
- The TX Power Level.
- The Service Class UUID for the iAP protocol, if the product has this service. For information about iAP, see [“The iPod Accessory Protocol”](#) (page 21).

During the Bluetooth discovery process the, Apple product prefers to display the Friendly Name of discovered Bluetooth accessories. Before the 2.1 version of the Bluetooth specification the Apple product would have to set up a connection to the accessory and do a Remote Name Request, which takes power, antenna time, and user's time. The Extended Inquiry Response feature, introduced in Bluetooth 2.1, lets an accessory send its Local Name and other information as part of the Inquiry Response and thereby increase the speed and efficiency of the discovery process.

Secure Simple Pairing

Every Bluetooth accessory that is compatible with an Apple product should:

- Use Secure Simple Pairing.
- Use the Numerical Comparison method if it has a display and input device supporting it.

Secure Simple Pairing greatly increases security and is a mandatory security feature introduced in the Bluetooth 2.1 specification. To protect against a man-in-the-middle attack, the Numerical Comparison association model should be used whenever feasible. See Volume 1, Section 5.4 in the Bluetooth Core Specification, Version 2.1 + EDR.

Profiles

The Apple knowledge base article support.apple.com/kb/ht3647 provides a complete list of the Bluetooth profiles that certain Apple products support. The Bluetooth specifications are the starting point for designing Bluetooth accessories that are compatible with these products. The following sections add information and requirements for some profiles, which can help accessory developers achieve superior results.

Device ID Profile (DID)

Every Bluetooth accessory that is compatible with an Apple product should:

- Support the Bluetooth Device ID Profile, version 1.3 or higher.
- Use the company identifier from the Assigned Numbers document assigned by the Bluetooth SIG as its Vendor ID value (VID). See www.bluetooth.org/Technical/AssignedNumbers/identifiers.htm (requires login). Bluetooth HID Profile accessories may use a VID assigned by the USB Implementers Forum (USB-IF at www.usb.org) if the manufacturer does not have a Bluetooth SIG company identifier.
- Use its VID value for the end product manufacturer.
- Use a ProductID value that uniquely identifies the product.
- Use a Version value that uniquely identifies the software version.

The Device ID profile lets the Apple product identify the implementation of the remote Bluetooth accessory. This is valuable information and can be used to bridge alternate interpretations of the Bluetooth specification when communicating with a remote accessory. Therefore it is important that the information in the Device ID record uniquely identify the implementation.

In the case of car kits, for instance, the same car kit might go into two different car models. Ideally the two car kits should have different Product IDs. However, it is acceptable for them to have the same ProductID as long as they have identical hardware, software, and features. If the implementations differ at all, they should have different Product IDs.

Hands-Free Profile (HFP)

If a Bluetooth accessory supports the Bluetooth Hands-Free Profile specification, it should be version 1.5 or higher.

Remote accessories can use the Bluetooth Hands-Free Profile for phone communications. To achieve the best user experience, the remote accessory should support the following features, which are optional in the Bluetooth specification.

Remote Audio Volume Control

Every Bluetooth accessory that is compatible with an Apple product and supports HFP should:

- Support Remote Audio Volume Control so the speaker volume on the Hands-Free accessory can be controlled from the Apple product as described in Section 4.28 in the Bluetooth Hands-Free Profile specification version 1.5.
- Set the Remote volume control bit in the Supported Features bitmap sent with the AT+BRSF= command.

In some situations it is easier for the user to control the output volume through the Apple product instead of directly on the remote accessory. For example, a passenger (or—if the car is parked—the driver) in a car could use the volume slider on the phone to control the audio volume. Volume control synchronization is outlined in Section 4.48.2 in the Bluetooth Hands-Free Profile specification version 1.5.

Indicator Event Reporting

Every Bluetooth accessory that is compatible with an Apple product and supports HFP should use indicator events reporting and not perform repetitive polling of status.

Apple products support all mandatory and optional indicators specified in HFP version 1.5 (service, call, callsetup, callheld, signal, roam, battchg). To minimize unnecessary polling of status using the AT+CIND? command, the remote accessory should enable indicator events reporting by sending an AT+CMER command. The Apple

product will then send a +CIEV event when there is a change in status of an indicator. The remote accessory should request the initial status using the AT+CIND=? and AT+CIND? commands, according to the HFP specification.

Voice Recognition Activation

Every Bluetooth accessory that is compatible with an Apple product and supports HFP should:

- Support Voice Recognition Activation, both AG and HF initiated as described in Section 4.25 in the Bluetooth Hands-Free Profile specification version 1.5.
- Set the “Voice recognition activation” bit in the “SupportedFeatures” bitmap sent with the AT+BRSF= command.

Apple products support voice recognition initiated by remote (Hands-Free) accessories and iOS (Audio Gateway) accessories.

Echo Cancellation and Noise Reduction

When echo cancellation and noise reduction are performed locally on a Hands-Free Bluetooth accessory, it should turn off echo cancellation and noise reduction on the Apple product by sending an AT+NREC command, as described in Section 4.24 in the Bluetooth Hands-Free Profile specification version 1.5.

Apple products support echo cancellation and noise reduction; these features are active by default. If a Hands-Free accessory also does echo cancellation and noise reduction it needs to turn these features off on the Apple product (the Audio Gateway). This avoids unnecessary degradation of audio quality due to double audio processing.

In-Band Ringing

Every Bluetooth accessory that is compatible with an Apple product and supports HFP should also support In-Band Ringing as specified in Section 4.13.1 in the Bluetooth Hands-Free Profile specification version 1.5. If the user sets a ring tone on the Apple product, the same ring tone should sound on the hands-free accessory.

Synchronous Connection

Every Bluetooth accessory that is compatible with an Apple product and supports HFP should:

- Support eSCO parameter set S2 and S3 and accept requests for these settings. See Section 5.6 of the Bluetooth *Hands-Free Profile* specification version 1.5.
- Request eSCO parameter set S2 or S3 when setting up a Synchronous Connection. Note that eSCO parameter set S1 should not be requested.

- Render audio within 40 ms after the SCO/eSCO connection has been set up.

The eSCO packet types offers retransmission of packets; traditional SCO packets are not retransmitted. This improves audio quality and the user's experience. The eSCO packet types 2-EV3 and 3-EV3 offer more time between packets, which can improve WiFi performance and allow for other concurrent Bluetooth connections.

Wide Band Speech

Every Bluetooth accessory that is compatible with an Apple product and supports HFP should support a Wide Band Speech Connection as described in Section 5.7.4 of the Bluetooth *Hands-Free Profile* specification version 1.6. If Wide Band Speech Connection is supported, it should support the T2 link parameter settings.

All iOS devices running iOS 5 or later support Wide Band Speech.

Message Access Profile (MAP)

Every Bluetooth accessory that is compatible with an Apple product and supports MAP should:

- Support Message Notification as described in Section 4.1 of the Bluetooth *Message Access Profile* specification, version 1.0.
- Register for notifications immediately after the connection is established, as described in Section 4.5 in the *Message Access Profile* specification, version 1.0.
- Not expect the TEL property to be present in the originator VCARD (the properties N and FN will be included). See Section 3.1.3 in the *Message Access Profile* specification, version 1.0.
- Not provide a user interface for sending messages. iOS devices do not support sending messages using MAP.

All iOS devices running iOS 6 or later support MAP.

Audio/Video Remote Control Profile (AVRCP)

To support the Audio/Video Remote Control Profile, a Bluetooth accessory that is compatible with an Apple product should support the buttons and operations listed in this section.

Supported Buttons

Every Bluetooth accessory that is compatible with an Apple product and supports the Audio/Video Remote Control Profile should use separate button commands to play and pause instead of toggling the play or pause state.

Supported Operations

Apple products support the following operation_IDs in Pass Through commands:

- Play
- Stop
- Pause
- Fast Forward
- Rewind
- Forward
- Backward

Repeat and Shuffle Modes

Every iOS device supports Repeat and Shuffle modes in the role of an AVRCP target. An AVRCP controller may use `SetPlayerApplicationSettingValue` to set a value on the iOS device and `GetPlayerApplicationSettingValue` to read a value, as described in Sections 6.5.4 and 6.4.3 of the Bluetooth *Audio/Video Remote Control Profile* specification version 1.4.

Notifications

Every Bluetooth accessory that is compatible with an Apple product and supports the AVRCP profile should register for notifications and not perform repetitive polling to determine the status of the Apple product.

Every iOS device supports registering for notifications in the role of an AVRCP Target, as described in Section 6.7 of the Bluetooth *Audio/Video Remote Control Profile* specification version 1.4. The commands `RegisterNotification` and `GetPlayStatus` are supported for these notifications:

- EVENT_PLAYBACK_STATUS_CHANGED
- EVENT_TRACK_CHANGED
- EVENT_NOW_PLAYING_CONTENT_CHANGED
- EVENT_AVAILABLE_PLAYERS_CHANGED
- EVENT_ADDRESSED_PLAYER_CHANGED
- EVENT_VOLUME_CHANGED

Volume Handling

Every Bluetooth accessory that is compatible with an Apple product and supports the AVRCP profile should support Absolute Volume, as described in Section 6.13 of the Bluetooth *Audio/Video Remote Control Profile* specification version 1.4.

Every iOS device supports volume handling in the role of AVRCP Controller.

Browsing

Every Bluetooth accessory that is compatible with an Apple product and supports Browsing (in controller role) as part of the AVRCP profile should:

- Not try to index or cache the entire library upon connection. The iOS device may contain tens of thousands of media items, each present multiple times in the hierarchy.
- When browsing a specific folder, do not fetch all its items. Only fetch those that are displayed to the user. It may prefetch a few items to improve the responsiveness of the user interface.
- Not reorder items (e.g. alphabetically).
- Not assume UIDs to be statically defined, especially in the root folder. The ordering and UIDs of folders and items may change at any point in future releases.
- Send the `SetBrowsedPlayer` command after receiving an `EVENT_UIDS_CHANGED` notification.
- Not assume that the UID passed to the `PlayItem` command will result in the media player playing that UID.

Currently only the built-in Music app supports browsing. When switching between players, an `EVENT_AVAILABLE_PLAYERS_CHANGED` notification and an `EVENT_ADDRESSED_PLAYER_CHANGED` notification will be generated. The UI then needs to look at the feature bit mask of the listed player to determine whether browsing is currently available.

All iOS devices running iOS 6 or later support AVRCP Browsing.

iOS App-Provided Metadata

An audio app running on an iOS device may use the iOS MediaPlayer Framework APIs to provide metadata about the current audio stream. The iOS device supplies this metadata to the accessory using AVRCP. For more information, see the `MPNowPlayingInfoCenter` class in Apple's MediaPlayer Framework documentation.

Advanced Audio Distribution Profile (A2DP)

Every Bluetooth accessory that is compatible with an Apple product and supports the Advanced Audio Distribution Profile should meet the requirements of the specification *Bluetooth Advanced Audio Distribution Profile*, Version 1.2. Additional Apple requirements are specified in this section.

SubBand Codec (SBC)

The SBC Codec Specific Information Elements, defined in Section 4.3.2 of the A2DP specification, that are applicable to Apple products are listed in [Table 1-1](#) (page 14).

Table 1-1 SubBand Codec Information Elements for Apple products

Element	Value
Sampling Frequency	44,100 Hz
Channel Mode	Stereo
Block Length	16
Subbands	8
Allocation Method	Loudness
Bitpool range	2 to 53. Bluetooth accessories for Apple products should support 53.

MPEG 2/4 AAC Codecs

iOS devices support the non-mandatory codec MPEG-2/4 AAC, as defined in Section 4.5 of the A2DP specification, Version 1.2. Accessories should use the AAC codec in addition to SBC, because it provides higher audio quality for a given bit rate.

Note: The following specifications provide details of Apple's implementation of the MPEG-2/4 AAC codec. In case of conflicts, the A2DP specification governs.

The MPEG 2/4 AAC Codec Specific Information Elements, defined in Section 4.5 of the A2DP specification, that are applicable to iOS devices are listed in [Table 1-2](#) (page 14).

Table 1-2 MPEG-2/4 AAC Codec Information Elements for iOS devices

Element	Value
Object Type	MPEG-2 AAC LC

Element	Value
Sampling Frequency	44,100 Hz
Channels	2
Bit rate	264,630 bps
VBR	0

AAC audio stream packets in iOS devices have the structure shown in [Table 1-3](#) (page 15).

Table 1-3 AAC audio packet for iOS devices

L2CAP Header	AVDTP Header	MPEG-4 LATM AudioMuxElement	MPEG-4 AAC Audio Payload
-----------------	-----------------	--------------------------------	-----------------------------

The AAC Media Payload Format, as defined in Section 4.5.4 of the A2DP specification, is formatted using LATM, as defined in Section 4 of IETF RFC 3016. The following notes apply to the packet fields shown in [Table 1-3](#) (page 15):

- The suggested L2CAP MTU value for each iOS device's AAC streaming channel is 885 bytes.
- The AVDTP Header is shown as the RTP header in Figure 4 of RFC 3016, and is the header defined in Section 7.2.1 of *Audio/Video Distribution Transport Protocol, Version 1.2*.
- The AudioMuxElement is the same as the RTP payload in RFC 3016. It is defined in Section 1.7.3, Table 1.32 in ISO/IEC 13818-3:2005, subpart 1. The muxConfigPresent argument to the AudioMuxElement is set to 1 (in-band mode), as recommended in Section 4.1 of RFC 3016. As recommended in Section 4.3 of RFC 3016, only one AudioMuxElement is put into each AVDTP packet.
- The audio payload is encoded using MPEG-4, as recommended in Section 4.5.4 of the A2DP specification.

Audio Routing

This section describes how an accessory can differentiate between various audio contents coming from an iOS device and use this information to decide playback behavior.

An accessory can receive audio data from the iOS device via either of two Bluetooth profiles:

- HFP using eSCO channel
- A2DP using ACL channel

The iOS device picks which channel to use depending on how the audio content is used. An audio path created for two way communication (such as phone calls or FaceTime) always uses the HFP (eSCO) route for sending audio data. Music and similar content uses the A2DP route. In the absence of a defined route, audio playback will default to the iOS device.

Audio Data Received via HFP Profile

Most of the audio content sent via HFP (eSCO) routes requires two way communication. Cases where HFP (eSCO) is used include (but are not limited to) cellular calls, FaceTime, and voice mail.

For any audio content that is being received via the HFP (eSCO) route, it is expected that both the speaker and the microphone of the accessory are dedicated to the Bluetooth link and should not handle any other audio content.

Audio Data Received via A2DP Profile

Audio content transferred via A2DP profiles can be broadly classified into two categories:

- Audio content from music, video, or game-like applications.
- System-generated sound for alerts and notifications.

Differentiating Audio Content from System Sounds

Music-like content can be differentiated from system sound by adding support for the AVRCP profile version 1.3 or later. The AVRCP profile allows an accessory to be aware of the audio playback state in the iOS device, using notifications.

When an iOS device initiates audio playback over an A2DP channel for playing music content, an AVRCP notification `EVENT_PLAYBACK_STATUS_CHANGED` is sent to indicate that playback status has changed to play state. See Section 6.7.2 of the AVRCP specification, version 1.4. This indicates that audio data via the A2DP profile contains music. When an iOS device initiates audio playback over an A2DP channel for playing system sound, an AVRCP notification is sent.

Expected Audio Routing Behavior for A2DP

The accessory should tune its audio routing behavior based on audio content over A2DP channel.

If audio data contains music, then it is expected that the accessory speakers are dedicated to audio data coming via the Bluetooth link and any other audio playback is paused. If audio data contains system sound, then it is expected that the accessory can render audio as desired. If the accessory is playing audio from a different source, then system sound data can be mixed with the existing track for playback; it is not necessary to pause existing audio playback on the device.

Bluetooth Low Energy

The Bluetooth 4.0 specification introduces Bluetooth Low Energy, a new wireless technology targeted for accessories with limited battery resources. If Bluetooth LE is supported, the accessory should follow the guidelines in this section.

Role

The Bluetooth accessory should implement either the Peripheral role as defined in the Bluetooth 4.0 specification, Volume 3, Part C, Section 2.2.2.3 or the Broadcaster role, as defined in Section 2.2.2.1.

Advertising Channels

The Bluetooth accessory should advertise on all three advertising channels (37, 38, and 39) at each advertising event. See the Bluetooth 4.0 specification, Volume 6, Part B, Section 4.4.2.1.

Advertising PDU

The Bluetooth accessory should use one of the following advertising PDUs:

- ADV_IND
- ADV_NOCONN_IND
- ADV_SCAN_IND

ADV_DIRECT_IND should not be used. See the Bluetooth 4.0 specification, Volume 6, Part B, Section 2.3.1.

Advertising Data

The advertising data sent by the Bluetooth accessory should contain at least the following information as described in the Bluetooth 4.0 specification, Volume 3, Part C, Section 11:

- Flags
- TX Power Level
- Local Name
- Services

The accessory may put the Local Name and the TX Power Level data in the SCAN_RSP PDU if, for example, it needs to reduce power consumption or not all of the advertising data fit into the advertising PDU. Note that, depending on its state, the Apple product may not always perform active scanning.

The primary services should always be advertised in the advertising PDU. Secondary services should not be advertised. Services not significant to the primary use case of the accessory may be omitted if space is limited in the Advertising PDU.

The advertising data and the scan response data in the SCAN_RSP PDU should comply with the formatting guidelines in the Bluetooth 4.0 specification, Volume 3, Part C, Section 18: it starts with a length field, followed by AD Type and AD Data.

Advertising Interval

The advertising interval of the Bluetooth accessory should be carefully considered, because it affects the time to discovery and connect performance. For a battery-powered accessory, its battery resources should also be considered.

To be discovered by the Apple product, the Bluetooth accessory should first use the recommended advertising interval of 20 ms for at least 30 seconds. If it is not discovered within the initial 30 seconds, the accessory may choose to save battery power and increase its advertising interval. Apple recommends using one of the following longer intervals to increase chances of discovery by the Apple product:

645 ms

768 ms

961 ms

1065 ms

1294 ms

Note: Longer advertising intervals usually result in longer discovery and connect times.

Connection Parameters

The Bluetooth accessory is responsible for the connection parameters used for the LE connection. The accessory should request connection parameters appropriate for its use case by sending an L2CAP Connection Parameter Update Request at the appropriate time. See the Bluetooth 4.0 specification, Volume 3, Part A, Section 4.20 for details.

The connection parameter request may be rejected if it does not comply with all of these rules:

Interval Max * (*Slave Latency* + 1) ≤ 2 seconds

Interval Min ≥ 20 ms

$Interval\ Min + 20\ ms \leq Interval\ Max$

$Slave\ Latency \leq 4$

$connSupervisionTimeout \leq 6\ seconds$

$Interval\ Max * (Slave\ Latency + 1) * 3 < connSupervisionTimeout$

The Apple product will not read or use the parameters in the Peripheral Preferred Connection Parameters characteristic. See the Bluetooth 4.0 specification, Volume 3, Part C, Section 12.5.

Privacy

The Bluetooth accessory should be able to resolve a Resolvable Private Address in all situations. Due to privacy concerns, the Apple product will use a Random Device Address as defined in the Bluetooth 4.0 specification, Volume 3, Part C, Section 10.8.

Permissions

The Bluetooth accessory should not require special permissions, such as pairing, authentication, or encryption to discover services and characteristics. It may require special permissions only for access to a characteristic value or a descriptor value. See the Bluetooth 4.0 specification, Volume 3, Part G, Section 8.1, fifth paragraph.

Pairing

The Bluetooth accessory should not request pairing. If, for security reasons, the accessory requires a bonded relationship with the Central, the Peripheral should reject the ATT request using the Insufficient Authentication error code, as appropriate. See the Bluetooth 4.0 specification, Volume 3, Part F, Section 4 for details.

As a result, the Apple product may proceed with the necessary security procedures.

Pairing may require user authorization depending on Apple product.

Services

Generic Access Profile Service

The Bluetooth accessory should implement the Device Name characteristic per the Bluetooth 4.0 specification, Volume 3, Part C, Section 12.1. The Device Name characteristic should be writeable.

Generic Attribute Profile Service

The Bluetooth accessory should implement the Service Changed characteristic only if the accessory has the ability to change its services during its lifetime.

The Apple product may use the Service Changed characteristic to determine if it can rely on previously read (cached) information from the device. See the Bluetooth 4.0 specification, Volume 3, Part G, Section 7.1.

Device Information Service

The Bluetooth accessory should implement the Device Information Service. The service UUID for this service should not be advertised in the Advertising Data. The following characteristics should be supported:

- Manufacturer Name String

- Model Number String

- Firmware Revision String

- Software Revision String

GATT Server

With iOS 6, applications may contribute services and characteristics to the GATT server that the iOS device makes available to the Bluetooth accessory. The recommendations in this section apply to the accessory in this case.

The iOS device implements the GAP Service Changed characteristic, because the database contents can change at any time. The Bluetooth accessory should therefore support the Characteristic Value Indication of this characteristic and, upon receiving indications, invalidate its database cache accordingly. See the Bluetooth 4.0 specification, Volume 3, Part G, Section 7.1.

The Bluetooth accessory should minimize the use of ATT/GATT requests and commands and only send what is necessary. For example, do not use GATT Discover All Services when the accessory is looking for specific services. Use Discover Primary Service By Service UUID instead. Less airtime equals less power consumption and better performance for both the accessory and the Apple device.

The Bluetooth accessory should be robust enough to handle any error gracefully. Pairing and Characteristic Value reads/writes may fail if the application that owns the service is not in the foreground and is not entitled to run in the background.

If an ATT Prepare Write Request is used, all queued attributes are contained within the same GATT Service.

Apple Protocols and Developer Support

This chapter describes Apple resources that can help you to get developer support for Apple-compatible Bluetooth accessory designs and to extend accessory capabilities beyond those supported by standard Bluetooth profiles.

Apple Bluetooth Development Mailing List

Questions or comments regarding Bluetooth accessory development for Apple products can be posted to the Apple mailing list for Bluetooth development, bluetooth-dev@lists.apple.com. To become a member of this mailing list, visit lists.apple.com/mailman/listinfo/bluetooth-dev.

Engineers on the Apple Bluetooth development team monitor this mailing list and will try to answer your questions. Please search the archives first, to see if your topic has already been discussed.

The iPod Accessory Protocol

Third-party accessories can use the iPod Accessory Protocol (iAP) to access advanced features of iOS devices. One such feature is the ability to communicate securely with third-party iOS applications via the iOS External Accessory Framework. For information about the External Accessory Framework, see [External Accessory Programming Topics](#) on the Apple iOS Developer site.

To incorporate iAP into an accessory design, the accessory developer must be a member of the Apple MFi licensing program and integrate specific MFi hardware into the accessory. For further information about MFi, see developer.apple.com/programs/mfi.

Headset Battery Level Indication

Any Hands-Free Bluetooth headset accessory can show its battery level to the user as an indicator icon in the iOS device status bar. This feature is supported on all iOS devices that support the Hands-Free Profile, including iPhone, iPod touch, and iPad.

Headset battery indication is implemented by two iOS-specific Bluetooth HFP AT commands, “[HFP Command AT+XAPL](#)” (page 22) and “[HFP Command AT+IPHONEACCEV](#)” (page 22)

HFP Command AT+XAPL

Description: Enables custom AT commands from a headset.

Initiator: Headset.

Format: AT+XAPL=[vendorID]-[productID]-[version],[features]

Parameters:

- **vendorID:** A string representation of the hex value of the vendor ID from the manufacturer, without the 0x prefix.
- **productID:** A string representation of the hex value of the product ID from the manufacturer, without the 0x prefix.
- **version:** The revision of the software.
- **features:** A base-10 representation of a bit field. Available features are:
 - 1 = the headset supports battery reporting.
 - 2 = the headset is docked or powered.
 - All other values are reserved.

Example: AT+XAPL=ABCD-1234-0100,3

Response: +XAPL=iPhone,[features]

HFP Command AT+IPHONEACCEV

Description: Reports a headset state change.

Initiator: Headset.

Format: AT+IPHONEACCEV=[*Number of key/value pairs*],[key1],[val1],[key2],[val2],...

Parameters:

- *Number of key/value pairs:* The number of parameters coming next.
- **key:** the type of change being reported.
 - 1 = battery level.
 - 2 = dock state.

- `val`: the value of the change.
 - Battery events: A string value between '0' and '9'.
 - Dock state: 0 = undocked, 1 = docked.

Example: `AT+IPHONEACCEV=1,1,3`

Document Revision History

This table describes the changes to *Bluetooth Accessory Design Guidelines for Apple Products*.

Date	Notes
2012-11-06	<p><i>Revision R6:</i></p> <p>Added section “Audio Routing” (page 15)</p> <p>Added section “Message Access Profile (MAP)” (page 11)</p> <p>Added section “GATT Server” (page 20)</p> <p>Added section “Browsing” (page 13)</p> <p>Removed 'USB' from vendor and product ID descriptions in “HFP Command AT+XAPL” (page 22).</p> <p>Made other corrections and updates.</p>
2011-10-14	<p><i>Revision R5:</i></p> <p>Moved “Headset Battery Level Indication” section to Apple Protocols chapter.</p> <p>Moved Low Energy chapter to Chapter 2.</p> <p>Added new section “Wide Band Speech” (page 11).</p> <p>Added new section “Apple Bluetooth Development Mailing List” (page 21).</p> <p>Added new section “Headset Battery Level Indication” (page 21).</p> <p>Added new material to section “Audio/Video Remote Control Profile (AVRCP)” (page 11).</p> <p>Clarified Remote Audio Volume Control and added Low Energy section.</p> <p>Made other corrections and updates.</p>

Date	Notes
2011-04-04	<i>Revision R4:</i> Changed specification language from requirements to guidelines.
2011-03-04	<i>Revision R3:</i> Added Mac as Apple products covered by this specification. Changed title of document from “Bluetooth Accessory Design Guidelines for iOS Devices” to “Bluetooth Accessory Design Guidelines for Apple Products.” Revised definition of Vendor ID source in “ Device ID Profile (DID) ” (page 8).
2010-11-30	<i>Revision R2:</i> Retitled document from <i>Designing Bluetooth Accessories for iOS Devices</i> . Corrected generic references to Apple devices, iOS devices, and iPod throughout the document; see “ IMPORTANT ” (page 5). Expanded specifications in section “ Advanced Audio Distribution Profile (A2DP) ” (page 14). Made minor other corrections and updates.
2010-09-08	<i>Revision R1:</i> First release.



Apple Inc.

© 2012 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 for personal use only and to print copies of documentation for personal 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 assist application developers to develop applications only for Apple-labeled computers.

Apple Inc.

1 Infinite Loop

Cupertino, CA 95014

408-996-1010

Apple, the Apple logo, FaceTime, iPad, iPhone, iPod, iPod touch, Mac, Numbers, and OS X 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.

Even though Apple has reviewed this document, 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 OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.