



Wireless Accessory Configuration Product Definition Specification

Release R2
September 10, 2014

Authors:
Apple



Apple Inc. © 2013-2014 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.

The Apple logo is a trademark of Apple Inc. Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws. 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.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Inc. 1 Infinite Loop Cupertino, CA 95014 408-996-1010

App Store is a service mark of Apple Inc.

iTunes Store is a registered service mark of Apple Inc.

Apple, the Apple logo, FireWire, iPhone, iPod, iPod classic, iPod nano, iPod shuffle, iPod touch, iTunes, Mac, Mac OS, Macintosh, Pages, and Spotlight are trademarks of Apple Inc., registered in the United States and other countries.

iPad, Numbers, and QuickStart are trademarks of Apple Inc.

IOS is a trademark or registered trademark of Cisco in the U.S. and other countries and is used under license.

Times is a registered trademark of Heidelberger Druckmaschinen AG, available from Linotype Library GmbH. Simultaneously published in the United States and Canada.

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.



Table of Contents

Introduction	4
Requirements	5
Wireless Accessory Configuration Setup Experience	7
Bonjour	10
Apple Device Information Element (IE)	11

Introduction

Apple's Wireless Accessory Configuration feature is designed to allow consumers to easily set up their wireless accessories with the network credentials already stored on their iOS or OS X device.

With a Wireless Accessory Configuration-enabled accessory, users can:

- Wirelessly discover their configurable accessory
- Give their accessory a descriptive name
- Pass network credentials to the accessory so it may join an infrastructure network
- Discover applications compatible with the accessory

Requirements

User requirements for using Wireless Accessory Configuration

Hardware

Wireless Accessory Configuration requires a device running iOS version 7.0 or greater or OS X 10.9 with AirPort Utility 6.3.1 or greater to act as the configuring device.

AND

A Wireless Accessory Configuration enabled accessory with Wi-Fi.

Network

Wireless Accessory Configuration requires a wireless TCP/IP network connection between the configuring device and accessory. The Wireless Accessory Configuration enabled accessory must be able to act as both a software Access Point (AP) and as a station (STA) device.

Target wireless networks require:

- an access point compatible with 802.11b/g, 802.11n, or 802.11ac

Wireless Accessory Configuration accessory requirements

All Wireless Accessory Configuration accessories must incorporate the following features and requirements.

Product features

Wireless Accessory Configuration accessories must include the following:

- an 802.11b/g, 802.11n, or 802.11ac radio module
- a status indicator to indicate when the device is in Wireless Accessory Configuration mode
 - It is strongly recommended that this status indicator only be used for indicating when the device is in Wireless Accessory Configuration mode, and not be used for any other purpose.
- the ability for the user to manually enter Wireless Accessory Configuration mode which optionally performs a full factory setting reset of the accessory
 - If a full factory reset is not performed, it is required that sensitive user information be erased. Examples of information to erase includes, but is not limited to:
 1. Passwords and authentication tokens (e.g. music service provider credentials, etc.)
 2. Medical data.
 3. Financial records.
 4. Personally identifiable information.

Additionally, it is strongly recommended that the accessory not use proprietary wireless technologies in the product that overlap with the international Wi-Fi spectrum.

Network requirements

All Wireless Accessory Configuration accessories must support a wireless network connection and Bonjour, Apple's service discovery protocol. Devices must also support changing the Bonjour name to a user defined value. The default name of all devices must be unique out of the box.

When in Software AP mode, the accessory must have a DHCP server as well as an HTTP server that supports persistent connections, i.e. multiple HTTP requests sent through the same TCP connection. When in STA mode, the accessory must support link-local IPv4 addressing as specified in RFC 3927.

Implementation requirements

All Wireless Accessory Configuration accessories must meet the following implementation requirements:

- incorporate the iPod Authentication Coprocessor
- meet all applicable requirements documented in the MFi Accessory Interface specification

Product behavioral requirements

All Wireless Accessory Configuration accessories must meet the following behavioral requirements:

- when an accessory is powered on and is unconfigured, it must automatically enter Wireless Accessory Configuration mode
- when an accessory has been in Wireless Accessory Configuration mode for more than 30 minutes, it must exit Wireless Accessory Configuration mode
- when entering Wireless Accessory Configuration mode, the software Access Point must use a unique SSID
- if an accessory's network credentials have been reset to factory setting defaults by the user, the accessory must fall back to the prescribed behaviors for an unconfigured device
- if an accessory has been configured, but loses its network connection, it must NOT enter Wireless Accessory Configuration mode automatically

Certification requirements

Wireless Accessory Configuration accessories must complete the following certification requirements:

- Wi-Fi Alliance "Wi-Fi certified" program
- MFi program certification

NOTE: Pre-existing MFi program certification on an accessory is not sufficient for Wireless Accessory Configuration accessory certification.

Wireless Accessory Configuration Setup Experience

The configuration process allows a configuring device, such as an iPhone, to send configuration information and network credentials to the accessory. This may include joining a Wi-Fi network, specifying a friendly name for the accessory, etc. The general flow of operation is:

2. Configuring device discovers accessories

Wi-Fi scans are conducted to find unconfigured accessories broadcasting the Apple information element (IE) in the accessory's software access point Wi-Fi beacon frames.

3. Configuring device joins accessory's temporary software access point network

4. Configuring device searches for accessory via Bonjour

Configuring device browses for `_mfi-config._tcp` and matches the accessory by its device ID.

The device ID in the Bonjour TXT record is the same one advertised as part of the Apple Device IE.

5. Configuring device resolves via Bonjour and connects via TCP to accessory

This performs normal Bonjour PTR→SRV→A/AAAA resolving then connects via TCP.

6. Configuring device authenticates accessory

For MFi-certified accessories, this performs MFi-SAP using the iPod Authentication Coprocessor.

Note: After this step, all subsequent control requests and responses are encrypted.

6. Configuring device builds config TLVs, encrypts it, and sends it in an HTTP request to accessory at `/config`

The TLV contains the information needed by the accessory to configure itself. See [Table 1. Configuration TLVs](#).

7. Accessory receives, validates, and saves config request

The accessory must not immediately apply the new configuration. Applying it at this point may interrupt its connection to the controller. The accessory must send its response, disable the sending side of its socket to signal it is done sending, then wait until it receives a FIN from the Configuring device (i.e. socket receive returns 0).

8. Configuring device receives accessory response and waits for the accessory to close its connection

The controller must disable the sending side of its socket to signal it has received the response then waits until it receives a FIN from the accessory (i.e. socket receive returns 0).

9. Accessory waits for controller to disconnect then deregisters its Bonjour service and increments its config seed ID

10. Accessory applies configuration and joins new Wi-Fi network

The accessory must wait until it has fully joined the new Wi-Fi network before continuing to the next step.

11. Accessory re-registers with Bonjour with the new config seed

It's important to only advertise the new configuration seed after the configuration has completed and the device has joined the new network. Otherwise, the controller may find a stale instance of the service and prematurely assume success.

12. Configuring device re-joins its original Wi-Fi network

13. Configuring device searches for accessory via Bonjour

The controller browses for `_mfi-config._tcp` and matches the accessory by its device ID (same one saved off from the earlier step).

14. Configuring device sends an HTTP request to `/configured` on the accessory to indicate configuration is complete

15. Configuring device reports success (or failure) to user

16. Configuring device prompts user to discover an application for the configured accessory if available

Table 1. Configuration TLVs.

Name	ID	Type	Description
"bundleSeedID"	0x01	String	Unique 10 character string assigned by Apple to an app via the Provisioning Portal (e.g. "24D4XFAF43").
"firmwareRevision"	0x02	String	Firmware revision of the accessory.
"hardwareRevision"	0x03	String	Hardware revision of the accessory.
"language"	0x04	String	BCP-47 language to configure the device for. See http://www.iana.org/assignments/language-subtag-registry .
"manufacturer"	0x05	String	Manufacturer of the accessory (e.g. "Apple").
"mfiProtocol"	0x06	String	Reverse-DNS string describing supported MFi accessory protocols (e.g. "com.acme.gadget") for accompanying applications. Note: there may be more than one of this item if multiple protocols are supported.

"model"	0x07	String	Model name of the device (e.g. Device1,1).
"name"	0x08	String	Name that accessory should use to advertise itself to the user.
"playPassword"	0x09	String	Password used to start an AirPlay stream to the accessory.
"serialNumber"	0x0A	String	Serial number of the accessory.
"wifiPSK"	0x0B	Data	Wi-Fi PSK for joining a WPA-protected Wi-Fi network. If it's between 8 and 63 bytes each being 32-126 decimal, inclusive then it's a pre-hashed password. Otherwise, it's expected to be a pre-hashed, 256-bit pre-shared key.
"wifiSSID"	0x0C	String	Wi-Fi SSID (network name) for the accessory to join. This should be UTF-8.

Bonjour

The Bonjour service type for Wireless Accessory Configuration is "_mfi-config._tcp". The name of the Bonjour service is the user-visible name of the accessory (e.g. "Basement Thermostat"). The name may contain any Unicode character and is encoded using UTF-8. It has a maximum length of 63 bytes (which may be fewer than 63 characters as a single Unicode character may require multiple bytes). Additional data needed for discovery-time metadata is advertised via a TXT record. This contains fields for feature detection, versions, etc.

Table 2. _mfi-config._tcp TXT record keys

Key	Description
"deviceid"	Globally unique ID for the accessory (e.g. the primary MAC address, such as "00:11:22:33:44:55").
"features"	Feature flag bits (e.g. "0x3" for bits 0 and 1). See Table 3. MFi Configuration Feature Flags .
"flags"	Status flags (e.g. 0x04 for bit 3). See Table 4. MFi Configuration Status Flags .
"protovers"	Protocol version string <major>.<minor> (e.g. "1.0"). Missing means 1.0.
"seed"	Configuration seed number. This is 0-255 and updates each time the software configuration changes.
"srcvers"	Source version number. Populated directly by the source code.

Table 3. MFi Configuration Feature Flags

Value	Bit	Description
0x00000001	0	App associated with this accessory.
0x00000004	2	Accessory supports TLV-based configuration.

Table 4. MFi Configuration Status Flags

Value	Bit	Description
0x01	0	Problem has been detected.
0x02	1	Device is not configured.

Apple Device Information Element (IE)

General Usage

This IE should be included in the following 802.11 management frames:

Probe response frames.

Beacon frames, if applicable.

Structure

This defines a vendor-specific 802.11 IE using the OUI 00-A0-40 (registered to Apple Inc.). The payload portion of the IE is composed of sub IEs defined by this document.

Table 5. Apple Device IE overall structure

Name	Size	Value	Description
Element ID	1	0xDD	Vendor specific element ID as specified in [REF2]
Length	1	Variable	Number of bytes in IE (excludes element ID and length bytes)
OUI	3	0x00 0xA0 0x40	Apple Inc. OUI reserved for this IE
Sub-type	1	0x00	Sub-type of the 00-A0-40 Apple Inc. OUI
Elements:	Variable	Variable	Sub IE elements defined by this spec

Table 6. Apple Device IE element structure

Name	Size	Description
Element ID	1	Vendor specific element ID as specified in [REF2]
Length	1	Number of bytes in the element payload (excludes element ID and length bytes)
Payload	Variable	Payload defined by the element ID

Payload

Table 7. Apple Device IE elements

Element ID	Name	Format	Description
0x00	Flags	n:bits	<p>Flags about the device: b0-b7, b8-b15, etc. See Table 8. Flags.</p> <p>Each flag is a bit. Bit numbering starts from the leftmost bit of the first byte and uses the minimum number of bytes needed to encode the bits. For example:</p> <p>If only bit 1 is set, it would be 0x40.</p> <p>If bit 1 (0x40) and bit 7 (0x01) are set, it would be 0x41.</p> <p>If bit 1 (0x40), bit 7 (0x01), and bit 10 (0x0020) are set, it would be 0x41, 0x20.</p> <p>If only bit 10 (0x0020), it would be 0x00, 0x20.</p>
0x01	Name	UTF-8	<p>Friendly name of the device.</p> <p>This should only be provided if the user configured a custom name or the firmware of the device has reason to believe it can provide a name that's better than the default name the client software will provide for it based on the model. Due to localization issues, it often better to only provide this element if the user has configured a name.</p>
0x02	Manufacturer	UTF-8	Machine-parsable manufacturer of the device (e.g. "Apple").
0x03	Model	UTF-8	Machine-parsable model of the device (e.g. "Device1,1").
0x04	OUI	3 bytes	OUI of the device including this IE.
0x05	dWDS	2 bytes	<1:DWDS Role><1:DWDS Flags>
0x06	Bluetooth MAC	6 bytes	MAC address of the Bluetooth radio, if applicable.

0x07	Device ID	6 bytes	<p>Globally unique ID of the device.</p> <p>This should be the primary MAC address of the device. If the device has multiple MAC addresses, one must be chosen as the primary MAC address such that it never changes (e.g. doesn't depend on the network interface currently active). The main purpose of this element is to allow devices to discover the device via Wi-Fi scans and then later associate it with an IP-based discovery method, such as Bonjour (where the device ID is expected to be reported via the TXT record).</p>
0x08-0xFF	TBD	TBD	Reserved
0xDD	Vendor-specific	n bytes	Same format as a normal vendor-specific IE element

Table 8. Flags

Value	Bit	Description
0x80	0	Supports AirPlay
0x40	1	Device is unconfigured
0x20	2	Supports MFi Configuration V1
0x10	3	Supports Wake on Wireless (WoW)
0x08	4	Device has interference robustness enabled
0x04	5	Device detected remote PPPoE server
0x02	6	Supports WPS
0x01	7	WPS is active on the device
0x008	8	Supports AirPrint
0x0008	12	Reserved
0x0002	14	Supports 2.4 GHz Wi-Fi networks
0x0001	15	Supports 5 GHz Wi-Fi networks
0x00008	16	Reserved
0x00004	17	Supports HomeKit Accessory Protocol

[REF2] Vendor-specific information element structure, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, IEEE Std. 802.11 - 2007