Introduction

Accessory interfaces
Accessory protocols
What’s new in iOS 8 and OS X Yosemite
Program information and wrap up
Accessory Interfaces

Wireless
- Bluetooth
- Wi-Fi

Wired
- Lightning connector
- USB
- Thunderbolt 2
- Headphone remote and mic
Bluetooth
## Bluetooth Low Energy

### Supported features

<table>
<thead>
<tr>
<th>Feature</th>
<th>iOS</th>
<th>OS X</th>
<th>App framework</th>
<th>For more information</th>
</tr>
</thead>
<tbody>
<tr>
<td>General communication with apps</td>
<td>✓</td>
<td>✓</td>
<td>Core Bluetooth</td>
<td>Bluetooth Design Guidelines</td>
</tr>
<tr>
<td>Apple Notification Center Service (ANCS)</td>
<td>✓</td>
<td></td>
<td>Notification Center</td>
<td>Bluetooth Design Guidelines</td>
</tr>
<tr>
<td>Hearing aid audio transport</td>
<td>✓</td>
<td></td>
<td>Core Audio</td>
<td>MFi Program</td>
</tr>
<tr>
<td>HealthKit</td>
<td>✓</td>
<td></td>
<td>HealthKit</td>
<td>Bluetooth Design Guidelines MFi Program</td>
</tr>
<tr>
<td>HomeKit</td>
<td>✓</td>
<td></td>
<td>HomeKit</td>
<td>MFi Program</td>
</tr>
<tr>
<td>iBeacon</td>
<td>✓</td>
<td></td>
<td>Core Location</td>
<td>iBeacon Licensing</td>
</tr>
</tbody>
</table>
## Bluetooth Low Energy

### Supported features

<table>
<thead>
<tr>
<th>Feature</th>
<th>iOS</th>
<th>OS X</th>
<th>App framework</th>
<th>For more information</th>
</tr>
</thead>
<tbody>
<tr>
<td>General communication with apps</td>
<td>✓</td>
<td>✓</td>
<td>Core Bluetooth</td>
<td>Bluetooth Design Guidelines</td>
</tr>
<tr>
<td>Apple Notification Center Service (ANCS)</td>
<td>✓</td>
<td></td>
<td>Notification Center</td>
<td>Bluetooth Design Guidelines</td>
</tr>
<tr>
<td>Hearing aid audio transport</td>
<td>✓</td>
<td></td>
<td>Core Audio</td>
<td>MFi Program</td>
</tr>
<tr>
<td>HealthKit</td>
<td>✓</td>
<td></td>
<td>HealthKit</td>
<td>Bluetooth Design Guidelines MFi Program</td>
</tr>
<tr>
<td>HomeKit</td>
<td>✓</td>
<td></td>
<td>HomeKit</td>
<td>MFi Program</td>
</tr>
<tr>
<td>iBeacon</td>
<td>✓</td>
<td></td>
<td>Core Location</td>
<td>iBeacon Licensing</td>
</tr>
</tbody>
</table>
## Bluetooth Low Energy

### Supported features

<table>
<thead>
<tr>
<th>Feature</th>
<th>iOS</th>
<th>OS X</th>
<th>App framework</th>
<th>For more information</th>
</tr>
</thead>
<tbody>
<tr>
<td>General communication with apps</td>
<td>✓</td>
<td>✓</td>
<td>Core Bluetooth</td>
<td>Bluetooth Design Guidelines</td>
</tr>
<tr>
<td>Apple Notification Center Service (ANCS)</td>
<td>✓</td>
<td></td>
<td>Notification Center</td>
<td>Bluetooth Design Guidelines</td>
</tr>
<tr>
<td>Hearing aid audio transport</td>
<td>✓</td>
<td></td>
<td>Core Audio</td>
<td>MFi Program</td>
</tr>
<tr>
<td>HealthKit</td>
<td>✓</td>
<td></td>
<td>HealthKit</td>
<td>Bluetooth Design Guidelines MFi Program</td>
</tr>
<tr>
<td>HomeKit</td>
<td>✓</td>
<td></td>
<td>HomeKit</td>
<td>MFi Program</td>
</tr>
<tr>
<td>iBeacon</td>
<td>✓</td>
<td></td>
<td>Core Location</td>
<td>iBeacon Licensing</td>
</tr>
</tbody>
</table>
### Classic Bluetooth

**Support for standard features**

<table>
<thead>
<tr>
<th>Feature</th>
<th>iOS</th>
<th>OS X</th>
<th>App framework</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced Audio Distribution Profile (A2DP)</td>
<td>✓</td>
<td>✓</td>
<td>Advanced Audio Distribution Profile (A2DP)</td>
</tr>
<tr>
<td>Audio/Video Remove Control Profile (AVRCP)</td>
<td>✓</td>
<td>✓</td>
<td>Audio/Video Remove Control Profile (AVRCP)</td>
</tr>
<tr>
<td>Dial Up Network (DUN) Profile</td>
<td>✓</td>
<td></td>
<td>Dial Up Network (DUN) Profile</td>
</tr>
<tr>
<td>File Transfer Profile (FTP)</td>
<td>✓</td>
<td></td>
<td>File Transfer Profile (FTP)</td>
</tr>
<tr>
<td>Hands-Free Profile (HFP)</td>
<td>✓</td>
<td></td>
<td>Hands-Free Profile (HFP)</td>
</tr>
<tr>
<td>Hardcopy Cable Replacement Profile (HCRP)</td>
<td>✓</td>
<td></td>
<td>Hardcopy Cable Replacement Profile (HCRP)</td>
</tr>
<tr>
<td>Headset Profile (HSP)</td>
<td>✓</td>
<td></td>
<td>Headset Profile (HSP)</td>
</tr>
<tr>
<td>Human Interface Device (HID) Profile</td>
<td>✓</td>
<td>✓</td>
<td>Human Interface Device (HID) Profile</td>
</tr>
<tr>
<td>Message Access Profile (MAP)</td>
<td>✓</td>
<td></td>
<td>Message Access Profile (MAP)</td>
</tr>
<tr>
<td>Object Push Profile (OPP)</td>
<td></td>
<td>✓</td>
<td>Object Push Profile (OPP)</td>
</tr>
<tr>
<td>Personal Area Network Profile (PAN)</td>
<td>✓</td>
<td></td>
<td>Personal Area Network Profile (PAN)</td>
</tr>
<tr>
<td>Phone Book Access Profile (PBAP)</td>
<td>✓</td>
<td></td>
<td>Phone Book Access Profile (PBAP)</td>
</tr>
<tr>
<td>Serial Port Profile (SPP)</td>
<td></td>
<td>✓</td>
<td>Serial Port Profile (SPP)</td>
</tr>
</tbody>
</table>
# Classic Bluetooth

**Support for extended features on iOS**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Headset battery level indication</td>
<td>Bluetooth Design Guidelines</td>
</tr>
<tr>
<td>Siri Eyes Free</td>
<td>Bluetooth Design Guidelines</td>
</tr>
<tr>
<td>iAP</td>
<td></td>
</tr>
<tr>
<td>• Access to media library</td>
<td></td>
</tr>
<tr>
<td>• Access to location</td>
<td></td>
</tr>
<tr>
<td>• Proprietary communication with apps</td>
<td>MFi Program</td>
</tr>
<tr>
<td>• Direct users to App Store when accessory is paired</td>
<td></td>
</tr>
<tr>
<td>• Game controllers</td>
<td></td>
</tr>
</tbody>
</table>

For more information
Wi-Fi

Wi-Fi Alliance certification
• 802.11a/b/g, 802.11n, 802.11ac
Dual-band operation
• 2.4 and 5 GHz
WPA2 security mode
Quality of services
• Wireless Multimedia (WMM)
IPv6
Lightning Connector

Lightning headphone module
• Richer controls, including iTunes Radio
• App-enabled headphones

New connector modules
All current Macs ship with USB 3 ports
All current Macs support 5 Gbps per port
900 mA per port for device power and charging
High performance storage devices should use UAS
Built-in driver support for XHCI 1.0 compliant host controllers
Thunderbolt 2

High speed data and 4K video on one cable
PCI Express and DisplayPort protocol
Dual-channel, bi-directional 20 Gbps per port
Up to 10W for bus-powered devices
Daisy chain up to six devices per port
Certification program ensures high quality devices

Refer to Thunderbolt Device Driver Programming Guide
Headphone Remote and Mic

Apps can receive remote control events
Standardized microphone
What's New in iOS 8 and OS X Yosemite
HealthKit Accessories

Terry Worley
Software Engineer, HealthKit team
HealthKit Sneak Peek

New framework in iOS 8
HealthKit Sneak Peek

New framework in iOS 8
Save and access health and fitness data
HealthKit Sneak Peek

New framework in iOS 8
Save and access health and fitness data
Class A data security
HealthKit Sneak Peek

New framework in iOS 8
Save and access health and fitness data
Class A data security
User privacy settings
HealthKit Sneak Peek

New framework in iOS 8
Save and access health and fitness data
Class A data security
User privacy settings
Searches and statistical queries
HealthKit Sneak Peek

New framework in iOS 8
Save and access health and fitness data
Class A data security
User privacy settings
Searches and statistical queries
Unit conversions
HealthKit Sneak Peek

New framework in iOS 8
Save and access health and fitness data
Class A data security
User privacy settings
Searches and statistical queries
Unit conversions
Notifications of new data
HealthKit Sneak Peek

New framework in iOS 8
Save and access health and fitness data
Class A data security
User privacy settings
Searches and statistical queries
Unit conversions
Notifications of new data
Integration with accessories
Health and Fitness Accessories
Health and Fitness Accessories

Bluetooth Low Energy is ideal for health and fitness
Health and Fitness Accessories

Bluetooth Low Energy is ideal for health and fitness
Accessories play the leading role in providing HealthKit data
Health and Fitness Accessories

Bluetooth Low Energy is ideal for health and fitness.
Accessories play the leading role in providing HealthKit data.
Accessories with built-in HealthKit support:
• Heart rate monitor
• Glucose sensor
• Blood pressure monitor
• Health thermometer
Health and Fitness Accessories

Bluetooth Low Energy is ideal for health and fitness
Accessories play the leading role in providing HealthKit data
Accessories with built-in HealthKit support
• Heart rate monitor
• Glucose sensor
• Blood pressure monitor
• Health thermometer
Once paired, HealthKit automatically controls the accessory
Developing Native HealthKit Accessories
Developing Native HealthKit Accessories

Adhere to Bluetooth Low Energy GATT Specifications
Developing Native HealthKit Accessories

Adhere to Bluetooth Low Energy GATT Specifications
Best practices
Developing Native HealthKit Accessories

Adhere to Bluetooth Low Energy GATT Specifications

Best practices

• Implement optional fields within the services, e.g. the heart rate field within the blood pressure service saved as its own data point in HealthKit
Developing Native HealthKit Accessories

Adhere to Bluetooth Low Energy GATT Specifications

Best practices

• Implement optional fields within the services, e.g. the heart rate field within the blood pressure service saved as its own data point in HealthKit

• Implement supported metadata, e.g. energy expended stored as metadata along with the heart rate measurement in HealthKit
Developing Native HealthKit Accessories

Adhere to Bluetooth Low Energy GATT Specifications

Best practices

- Implement optional fields within the services, e.g. the heart rate field within the blood pressure service saved as its own data point in HealthKit
- Implement supported metadata, e.g. energy expended stored as metadata along with the heart rate measurement in HealthKit
- Implement optional characteristics, e.g. sensor contact to help identify valid data
Developing Native HealthKit Accessories

Adhere to Bluetooth Low Energy GATT Specifications

Best practices

• Implement optional fields within the services, e.g. the heart rate field within the blood pressure service saved as its own data point in HealthKit

• Implement supported metadata, e.g. energy expended stored as metadata along with the heart rate measurement in HealthKit

• Implement optional characteristics, e.g. sensor contact to help identify valid data

• Implement optional services, e.g. battery Service
Developing Other Accessories
Developing Other Accessories

Custom solutions can contribute to HealthKit
Developing Other Accessories

Custom solutions can contribute to HealthKit

Use any service that makes sense for your accessory. That might include the ExternalAccessory framework, CoreBluetooth, USB, or Wi-Fi
Developing Other Accessories

Custom solutions can contribute to HealthKit

Use any service that makes sense for your accessory. That might include the ExternalAccessory framework, CoreBluetooth, USB, or Wi-Fi

Then use the HealthKit framework APIs from within your app to add your data
Custom solutions can contribute to HealthKit

Use any service that makes sense for your accessory. That might include the ExternalAccessory framework, CoreBluetooth, USB, or Wi-Fi

Then use the HealthKit framework APIs from within your app to add your data

We hope these custom solutions will join in
HealthKit Software Support
@property (nonatomic) HKHealthStore *healthStore;
[self setHealthStore:[[HKHealthStore alloc] init]];
@property (nonatomic) HKHealthStore *healthStore;
[self setHealthStore:=[[HKHealthStore alloc] init]];
...
HKQuantityType *heartRateType =
    [HKObjectType quantityTypeForIdentifier:HKQuantityTypeIdentifierHeartRate];
@property (nonatomic) HKHealthStore *healthStore;
[self setHealthStore:[[HKHealthStore alloc] init]];
...
HKQuantityType *heartRateType =
[HKObjectType quantityTypeForIdentifier:HKQuantityTypeIdentifierHeartRate];
@property (nonatomic) HKHealthStore *healthStore;
[self setHealthStore:[[HKHealthStore alloc] init]];
...
HKQuantityType *heartRateType =
   [HKObjectType quantityTypeForIdentifier:HKQuantityTypeIdentifierHeartRate];
HKObserverQuery *heartRateQuery =
   [[HKObserverQuery alloc] initWithSampleType:heartRateType predicate:nil
    updateHandler:^(HKObserverQuery *query,
          void (^completion)(void), NSError *error) {

    }];
@property (nonatomic) HKHealthStore *healthStore;
[self setHealthStore:=[[HKHealthStore alloc] init]];
...
HKQuantityType *heartRateType =
    [HKObjectType quantityTypeForIdentifier:HKQuantityTypeIdentifierHeartRate];
HKObserverQuery *heartRateQuery =
    [[HKObserverQuery alloc] initWithSampleType:heartRateType predicate:nil
    updateHandler:^(HKObserverQuery *query,
                      void (^completion)(void), NSError *error) {
        if (error == nil) {
            // Use HealthKit to fetch the new sample(s) from the database.
            [self fetchNewSamples];
        }
    }];
@property (nonatomic) HKHealthStore *healthStore;
[self setHealthStore:[[HKHealthStore alloc] init]];
...

HKQuantityType *heartRateType =
[HKObjectType quantityTypeForIdentifier:HKQuantityTypeIdentifierHeartRate];
HKObserverQuery *heartRateQuery =
[[HKObserverQuery alloc] initWithSampleType:heartRateType predicate:nil
updateHandler:^(HKObserverQuery *query,
    void (^completion)(void), NSError *error) {
    if (error == nil) {
        // Use HealthKit to fetch the new sample(s) from the database.
        [self fetchNewSamples];
    }
    }];

[self.healthStore executeQuery:heartRateQuery];
Demo
Heart rate monitor

Jorge Moriñigo
Software Engineer, HealthKit team
## Related Sessions

<table>
<thead>
<tr>
<th>Session</th>
<th>Location</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introducing HealthKit</td>
<td>Mission</td>
<td>Tuesday 10:15AM</td>
</tr>
<tr>
<td>Health and Fitness Get Together</td>
<td>Folsom</td>
<td>Tuesday 4:30PM</td>
</tr>
</tbody>
</table>
HomeKit Accessories

Kevin McLaughlin
Wireless Software Engineering
HomeKit
HomeKit

Bring exciting new accessories to our users
HomeKit

Bring exciting new accessories to our users
Provide consistent user experience
HomeKit

Bring exciting new accessories to our users
Provide consistent user experience
Learn about APIs at HomeKit session
HomeKit Accessory Protocol
HomeKit Accessory Protocol
HomeKit Accessory Protocol

HomeKit
Accessory
Protocol
“HAP”
HomeKit Accessory Protocol
Connecting accessories to HomeKit
HomeKit Accessory Protocol
Connecting accessories to HomeKit

Transports
- Bluetooth Low Energy
- IP
HomeKit Accessory Protocol

Connecting accessories to HomeKit

Transports
- Bluetooth Low Energy
- IP

Security
- Bi-directional authentication
- Per-session encryption
HomeKit Accessory Protocol

Connecting accessories to HomeKit

Transports
- Bluetooth Low Energy
- IP

Security
- Bi-directional authentication
- Per-session encryption

Common functionality definitions
- Services
- Characteristics
HomeKit Accessory Profiles

Services
- Garage door openers
- Lights
- Door locks
- Thermostats
- IP camera controls
- Switches
- ...
- Custom

Characteristics
- Power state
- Lock state
- Target state
- Brightness
- Model number
- Current temperature
- ...
- Custom
HomeKit Accessory Profiles

Services
- Garage door openers
- Lights
- Door locks
- Thermostats
- IP camera controls
- Switches
- ...
- Custom

Characteristics
- Power state
- Lock state
- Target state
- Brightness
- Model number
- Current temperature
- ...
- Custom
HomeKit Accessory Protocol Layers
HomeKit Accessory Protocol Layers

- Bluetooth LE
- IP
HomeKit Accessory Protocol Layers

- Bluetooth LE
- L2CAP
- TCP
- IP
HomeKit Accessory Protocol Layers

- Attribute Protocol (ATT)
- L2CAP
- Bluetooth LE
- HTTP
- TCP
- IP
HomeKit Accessory Protocol Layers

- Bluetooth LE
- L2CAP
- Attribute Protocol (ATT)
- Generic Attribute Profile (GATT)

Protocol Layers:
- IP
- TCP
- HTTP
- JSON
HomeKit Accessory Protocol Layers

HomeKit Accessory Protocol

Generic Attribute Profile (GATT)

Attribute Protocol (ATT)

L2CAP

Bluetooth LE

JSON

HTTP

TCP

IP
HomeKit Accessory Protocol Layers

- HomeKit
- HomeKit Accessory Protocol
  - Generic Attribute Profile (GATT)
  - Attribute Protocol (ATT)
  - L2CAP
  - Bluetooth LE
  - JSON
  - HTTP
  - TCP
  - IP
Garage Door Opener

Services and characteristics example
Garage Door Opener

Services and characteristics example

accessory : {
Garage Door Opener
Services and characteristics example

accessory : {
    service1 : “public.hap.accessory-information” {

Garage Door Opener
Services and characteristics example

accessory : {
    service1 : "public.hap.accessory-information" {
        characteristic : "serial-number"
Garage Door Opener

Services and characteristics example

accessory : {
    service1 : “public.hap.accessory-information” {
        characteristic : “serial-number”
        characteristic : “identify”
    }
}
Garage Door Opener
Services and characteristics example

accessory : {
  service1 : “public.hap.accessory-information” {
    characteristic : “serial-number”
    characteristic : “identify”
  }
  service2 : “public.hap.garage-door-opener” {

Garage Door Opener

Services and characteristics example

accessory : {
    service1 : “public.hap.accessory-information” {
        characteristic : “serial-number”
        characteristic : “identify”
    }
    service2 : “public.hap.garage-door-opener” {
        characteristic : “target-state”
    }
}
Garage Door Opener
Services and characteristics example

accessory : {
    service1 : “public.hap.accessory-information” {
        characteristic : “serial-number”
        characteristic : “identify”
    }
    service2 : “public.hap.garage-door-opener” {
        characteristic : “target-state”
        characteristic : “current-state”
    }
}
Garage Door Opener

Services and characteristics example

```json
accessory : {
    service1 : "public.hap.accessory-information" {
        characteristic : "serial-number"
        characteristic : "identify"
    }
    service2 : "public.hap.garage-door-opener" {
        characteristic : "target-state"
        characteristic : "current-state"
        characteristic : "obstruction-detected"
    }
}
```
accessory : {
    service1 : "public.hap.accessory-information" {
        characteristic : "serial-number"
        characteristic : "identify"
    }
    service2 : "public.hap.garage-door-opener" {
        characteristic : "target-state"
        characteristic : "current-state"
        characteristic : "obstruction-detected"
    }
    service3 : "public.hap.lightbulb" {

Garage Door Opener
Services and characteristics example

```json
accessory: {
    service1: "public.hap.accessory-information" { 
        characteristic: "serial-number"
        characteristic: "identify"
    }
    service2: "public.hap.garage-door-opener" { 
        characteristic: "target-state"
        characteristic: "current-state"
        characteristic: "obstruction-detected"
    }
    service3: "public.hap.lightbulb" { 
        characteristic: "on"
    }
}
```
IP Details

Remote access through HomeKit
IP Details

Remote access through HomeKit
Bonjour for accessory discovery
IP Details

Remote access through HomeKit

Bonjour for accessory discovery

A single IP accessory may present multiple “accessories”
  • Enables bridges
  • Bridge is responsible for translating to and from HAP
IP Details

Remote access through HomeKit

Bonjour for accessory discovery

A single IP accessory may present multiple “accessories”
• Enables bridges
• Bridge is responsible for translating to and from HAP

RESTful API to interact with accessories, services, and characteristics
Bluetooth Low Energy Details
Remote access through HomeKit
Bluetooth Low Energy Details

Remote access through HomeKit
Apple-defined advertisement data for HAP
Bluetooth Low Energy Details

Remote access through HomeKit
Apple-defined advertisement data for HAP
Accessories are not identifiable from their advertisement data
Bluetooth Low Energy Details

Remote access through HomeKit
Apple-defined advertisement data for HAP
Accessories are not identifiable from their advertisement data
HAP security instead of Bluetooth Low Energy pairing
Protocol Security

- End-to-end encryption
- Initial setup secured directly between iOS and accessory
- Perfect forward secrecy
- Standard cryptography
## Security Details

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secure Remote Password (SRP)</td>
<td>Encrypts and authenticates initial pairing key exchange</td>
</tr>
<tr>
<td>Ed25519</td>
<td>Long-term keys for pairing and authentication</td>
</tr>
<tr>
<td>Curve25519</td>
<td>Encrypts initial authentication for each session</td>
</tr>
<tr>
<td>HKDF-SHA-512</td>
<td>Per-session ephemeral encryption key derivation</td>
</tr>
<tr>
<td>ChaCha20-Poly1305</td>
<td>Encrypts and authenticates HAP data</td>
</tr>
</tbody>
</table>

Defined in pairing specification

Support available at HomeKit lab and through MFi program
Next Steps
Next Steps

MFi program

- Access to HomeKit specifications for protocol, pairing, and profiles
- Development and certification tool
- Available soon
Next Steps

MFi program
- Access to HomeKit specifications for protocol, pairing, and profiles
- Development and certification tool
- Available soon

Learn HomeKit APIs
- Attend HomeKit session and lab
AirPlay
AirPlay

Audio handling changes
• Accessory side skew compensation
• Buffer underrun handling
AirPlay

Audio handling changes
• Accessory side skew compensation
• Buffer underrun handling

New reference platform
• BSD on Raspberry Pi
iAP2 and External Accessory Framework
iAP2 and External Accessory Framework
Media playback enhancements for accessories
iAP2 and External Accessory Framework

Media playback enhancements for accessories

Access to playback queue
iAP2 and External Accessory Framework
Media playback enhancements for accessories

Access to playback queue
Play all tracks
iAP2 and External Accessory Framework

Media playback enhancements for accessories

Access to playback queue
Play all tracks
Seek to specific time in music track
iAP2 and External Accessory Framework

Media playback enhancements for accessories

Access to playback queue
Play all tracks
Seek to specific time in music track
Get non-localized bundle identifier for media apps
iAP2 and External Accessory Framework

Media playback enhancements for accessories

- Access to playback queue
- Play all tracks
- Seek to specific time in music track
- Get non-localized bundle identifier for media apps
- Get progress information during media library synchronization
iAP2 and External Accessory Framework

Additional enhancements for accessories
iAP2 and External Accessory Framework

Additional enhancements for accessories

Simplified time synchronization
iAP2 and External Accessory Framework

Additional enhancements for accessories

Simplified time synchronization
Get detailed charge state information
iAP2 and External Accessory Framework

Additional enhancements for accessories

Simplified time synchronization
Get detailed charge state information
Smaller footprint link layer reference code
iAP2 and External Accessory Framework

Additional enhancements for accessories

Simplified time synchronization
Get detailed charge state information
Smaller footprint link layer reference code
Multiple apps can open sessions for the same External Accessory protocol simultaneously
Wireless Accessory Configuration
Wireless Accessory Configuration

Configure Wi-Fi accessories from within your app

Import External Accessory framework
Implement `EAWiFiUnconfiguredAccessoryBrowserDelegate` protocol
Instantiate `EAWiFiUnconfiguredAccessoryBrowser`
Search for unconfigured Wi-Fi accessories

Search for all unconfigured Wi-Fi accessories
[browser startSearchingForUnconfiguredAccessoriesMatchingPredicate:nil];

or filter for a subset of accessories
[browser startSearchingForUnconfiguredAccessoriesMatchingPredicate:
    [NSPredicate predicateWithFormat:@"name = 'Amalgamated Audio'"]];
Wireless Accessory Configuration
Delegate callbacks

New accessories were found
- (void)accessoryBrowser:(EAWiFiUnconfiguredAccessoryBrowser*)
  browser didFindUnconfiguredAccessories:(NSSet*)accessories

Previously found accessory is no longer available
- (void)accessoryBrowser:(EAWiFiUnconfiguredAccessoryBrowser*)
  browser didRemoveUnconfiguredAccessories:(NSSet*)accessories

State of the browser has changed
- (void)accessoryBrowser:(EAWiFiUnconfiguredAccessoryBrowser*)
  browser didUpdateState:(EAWiFiUnconfiguredAccessoryBrowserState)state
Wireless Accessory Configuration

User selects an accessory
Wireless Accessory Configuration

User selects an accessory

Begin configuring the accessory

```objective-c
[browser configureAccessory:accessoryToConfigure
     withConfigurationUIOnViewController:self];
```
Wireless Accessory Configuration
User configures Wi-Fi accessory
Wireless Accessory Configuration

User configures Wi-Fi accessory
Wireless Accessory Configuration

User configures Wi-Fi accessory
WirelessAccessoryConfiguration
Configuration complete
Summary

HealthKit
HomeKit
Media playback enhancements for accessories
App-driven Wireless Accessory Configuration
References

MFi Program
http://developer.apple.com/mfi/

iBeacon

Bluetooth Accessory Design Guidelines for Apple Products
http://developer.apple.com/bluetooth/

Case Design Guidelines for Apple Devices
http://developer.apple.com/resources/cases/
More Information

Stephen Chick
iPhone Evangelist
crack@apple.com

Craig Keithley
MFi and I/O Technologies Evangelist
keithley@apple.com

Mark Tozer
Desktop Technologies Evangelist
tozer@apple.com

Apple Developer Forums
http://devforums.apple.com
# Related Sessions

<table>
<thead>
<tr>
<th>Session</th>
<th>Location</th>
<th>Date and Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introducing HealthKit</td>
<td>Mission</td>
<td>Tuesday 10:15AM</td>
</tr>
<tr>
<td>What’s New in Core Location</td>
<td>Marina</td>
<td>Tuesday 2:00PM</td>
</tr>
<tr>
<td>Introducing HomeKit</td>
<td>Marina</td>
<td>Tuesday 4:30PM</td>
</tr>
<tr>
<td>Adopting AirPrint</td>
<td>Pacific Heights</td>
<td>Friday 9:00AM</td>
</tr>
<tr>
<td>Designing for Game Controllers</td>
<td>Mission</td>
<td>Friday 10:15AM</td>
</tr>
<tr>
<td>Labs</td>
<td>Location</td>
<td>Time</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>I/O Technologies Lab</td>
<td>Core OS Lab A</td>
<td>Tuesday 9:00AM</td>
</tr>
<tr>
<td>Accessories and I/O Technologies Lab</td>
<td>Core OS Lab A</td>
<td>Tuesday 11:30AM</td>
</tr>
<tr>
<td>I/O Technologies Lab</td>
<td>Core OS Lab A</td>
<td>Thursday 11:30AM</td>
</tr>
<tr>
<td>Accessories and I/O Technologies Lab</td>
<td>Core OS Lab A</td>
<td>Thursday 12:45PM</td>
</tr>
<tr>
<td>HomeKit Lab</td>
<td>Services Lab A</td>
<td>Thursday 12:45PM</td>
</tr>
<tr>
<td>HealthKit Lab</td>
<td>Services Lab B</td>
<td>Friday 9:00AM</td>
</tr>
</tbody>
</table>