What’s New in Core Bluetooth

Duy Phan, Bluetooth Engineer
Yilok Wong, Bluetooth Engineer
Meghna Lav, Bluetooth Engineer
Introduction
Low energy 2 Mbps
Advertising extensions
Core Bluetooth for BR/EDR
Core Bluetooth dual-mode
User privacy
PacketLogger
Summary
Introduction

Duy Phan, Bluetooth Engineer
Low Energy 2 Mbps
LE 2 Mbps
LE 2 Mbps

New feature in Bluetooth 5.0
LE 2 Mbps

New feature in Bluetooth 5.0

Physical layer rate increased from 1 Mbps to 2 Mbps
New feature in Bluetooth 5.0

Physical layer rate increased from 1 Mbps to 2 Mbps
LE 2 Mbps

New feature in Bluetooth 5.0

Physical layer rate increased from 1 Mbps to 2 Mbps
LE 2 Mbps

New feature in Bluetooth 5.0

Physical layer rate increased from 1 Mbps to 2 Mbps

Faster and more power efficient connection
LE 2 Mbps

New feature in Bluetooth 5.0

Physical layer rate increased from 1 Mbps to 2 Mbps

Faster and more power efficient connection

Transparent to the application
LE 2 Mbps

New feature in Bluetooth 5.0
Physical layer rate increased from 1 Mbps to 2 Mbps
Faster and more power efficient connection
Transparent to the application
Accessories must support LE 2 Mbps
LE 2 Mbps

New feature in Bluetooth 5.0

Physical layer rate increased from 1 Mbps to 2 Mbps

Faster and more power efficient connection

Transparent to the application

Accessories must support LE 2 Mbps

Available starting with iPhone 8, Apple TV 4K, and Apple Watch Series 4
LE 2 Mbps Throughput (kbps)

- Write With Response: 2.5
- Write Without Response: 37
- +Larger MTU: 48
- +Extended Data Length: 135
- +LE L2CAP: 197
- +15ms Interval: 394
- +LE 2Mbps: 670
Advertising Extensions
Advertising Extensions

Adv 31

ExtAdv

1 Mbps

Adv 31

ExtAdv

1 Mbps

Adv 31

ExtAdv

1 Mbps
Advertising Extensions

ADV

ExtADV

31

1 Mbps

255

1 Mbps

2 Mbps

NEW
Extended Scan

Scans for extended advertisements
Extended Scan

Scans for extended advertisements

Accessories must support extended advertisements with LE 2 Mbps
Extended Scan

Scans for extended advertisements

Accessories must support extended advertisements with LE 2 Mbps

Support extended advertisement payload up to 124 bytes
Extended Scan

Scans for extended advertisements

Accessories must support extended advertisements with LE 2 Mbps

Support extended advertisement payload up to 124 bytes

4 times the advertisement data that an accessory can send today
Extended Scan

Scans for extended advertisements

Accessories must support extended advertisements with LE 2 Mbps

Support extended advertisement payload up to 124 bytes

4 times the advertisement data that an accessory can send today

Transparent to application
Extended Scan

Scans for extended advertisements

Accessories must support extended advertisements with LE 2 Mbps

Support extended advertisement payload up to 124 bytes

4 times the advertisement data that an accessory can send today

Transparent to application

New API to query for platform support

```swift
class func supports(_ features: CBCentralManager.Feature) -> Bool
static var extendedScanAndConnect: CBCentralManager.Feature { get }
```
Extended Connections
Extended Connections

Supports connections to connectable extended advertisements
Extended Connections

Supports connections to connectable extended advertisements

Improves existing connection exchange protocol
Legacy Connections

Adverser

Scanner

NEW
Legacy Connections

Advertiser

Scanner

NEW
Legacy Connections

Advertiser

Scanner

ADV

X No ACK

Conn_Indication

NEW
Extended Connections

Advertiser

ExtADV

Advertiser

Scanner
Extended Connections

Advertiser

ExtADV

Advertiser

Scanner

Connection_Request
Extended Connections

Advertiser

ExtADV

Connection_Request

Advertiser

Connection_Response

Scanner

Connection_Request
Extended Connections

Advertiser

ExtADV

Connection_Request

Scanner

Connection_Response

Wakes Host Processor
Extended Connections

Advertiser

ExtADV

Connection_Request

Scanner

Connection_Response

Wakes Host Processor

2 Mbps
Extended Connections

Supports connections to peripheral connectable extended advertisements

Improves on connection protocol exchange

More robust and power efficient
Extended Connections

Supports connections to peripheral connectable extended advertisements

Improves on connection protocol exchange

More robust and power efficient

Transparent to application
Extended Connections

Supports connections to peripheral connectable extended advertisements

Improves on connection protocol exchange

More robust and power efficient

Transparent to application

Accessories must support connectable extended advertisements
Extended Connections

Supports connections to peripheral connectable extended advertisements

Improves on connection protocol exchange

More robust and power efficient

Transparent to application

Accessories must support connectable extended advertisements

New API to query for platform support

```swift
class func supports(_ features: CBCentralManager.Feature) -> Bool
static var extendedScanAndConnect: CBCentralManager.Feature { get }
```
Core Bluetooth for BR/EDR

Yilok Wong, Bluetooth Engineer
Core Bluetooth 2018

Profiles
L2CAP
BR/EDR

Core Bluetooth
GATT
ATT
L2CAP
Low Energy
Core Bluetooth BR/EDR

NEW
Core Bluetooth BR/EDR

Use Core Bluetooth with BR/EDR Bluetooth devices
Core Bluetooth BR/EDR

Use Core Bluetooth with BR/EDR Bluetooth devices

Bluetooth SIG GATT protocol running over BR/EDR
Core Bluetooth BR/EDR

Use Core Bluetooth with BR/EDR Bluetooth devices

Bluetooth SIG GATT protocol running over BR/EDR

Same CBPeripheral APIs
Core Bluetooth BR/EDR

Use Core Bluetooth with BR/EDR Bluetooth devices

Bluetooth SIG GATT protocol running over BR/EDR

Same CBPeripheral APIs

New API in CBCentralManager
Core Bluetooth BR/EDR

Use Core Bluetooth with BR/EDR Bluetooth devices

Bluetooth SIG GATT protocol running over BR/EDR

Same CBPeripheral APIs

New API in CBCentralManager

Available today with latest iOS, watchOS, and tvOS
Core Bluetooth BR/EDR

Use Core Bluetooth with BR/EDR Bluetooth devices

Bluetooth SIG GATT protocol running over BR/EDR

Same CBPeripheral APIs

New API in CBCentralManager

Available today with latest iOS, watchOS, and tvOS

Add support to your accessory
Registering for Connection Events
Registering for Connection Events

Connection registration by the Central
• Register by Service
• Register by Peripheral
Registering for Connection Events

Connection registration by the Central

• Register by Service
• Register by Peripheral

```swift
open class CBCentralManager : CBManager {
    open func registerForConnectionEvents(options:[CBConnectionEventMatchingOption:Any]?)
}
```
Registering for Connection Events

Connection registration by the Central

• Register by Service
• Register by Peripheral

```swift
open class CBCentralManager : CBManager {
    open func registerForConnectionEvents(options:[CBConnectionEventMatchingOption:Any]?)
}
```

```swift
extension CBConnectionEventMatchingOption {
    public static let serviceUUIDs: CBConnectionEventMatchingOption
    public static let peripheralUUIDs: CBConnectionEventMatchingOption
}
```
Connection Event
Connection Event

Delegate callback

• Sent on matching connection

• Sent after registration if a matching connection already established
Connection Event

Delegate callback

• Sent on matching connection
• Sent after registration if a matching connection already established

```swift
optional func centralManager(_ central: CBCentralManager, connectionEventDidOccur event: CBConnectionEvent, for peripheral: CBPeripheral)
```
Incoming Connection
private var central: CBCentralManager!
central = CBCentralManager(delegate: self, queue: nil)
let matchingOptions = [CBCConnectionEventMatchingOption.serviceUUIDs : [myServiceUUID]]

central.registerForConnectionEvents(options: matchingOptions)
Discover
Discover

Inquiry Scan

Inquiry Response
Discover

Inquiry Scan

Inquiry Response
Connect/Pair

Bluetooth

To pair an Apple Watch with your iPhone, go to the Watch app.
Connect/Pair

- Bluetooth is turned on.
- A Bluetooth pairing request is displayed, asking to enter the code 846031 and select "Pair".
- A headphone icon is shown on the right side of the page.
Connect/Pair

BR/EDR Connected and Paired
Delegate Callback
Delegate Callback

// Connection Event
func centralManager(_ central: CBCentralManager,
connectionEventDidOccur event: CBConnectionEvent,
for peripheral: CBPeripheral) {

}
Delegate Callback

```swift
func centralManager(_ central: CBCentralManager, connectionEventDidOccur event: CBConnectionEvent, for peripheral: CBPeripheral) {

    // Handle connection event

}
```
// Initialization
private var cbManager: CBCentralManager!

cbManager = CBCentralManager(delegate: self, queue: nil)

// Registering for Gatt Connection events
let matchingOptions = [CBConnectionEventMatchingOption.serviceUUIDs : [myServiceUUID]]()

cbManager.registerForConnectionEvents(options: matchingOptions)

// Delegate callback

func centralManager(_ central: CBCentralManager, connectionEventDidOccur event: CBConnectionEvent, for peripheral: CBPeripheral) {
    switch event {
    case .peerConnected:
        // We are interested in this peripheral, clear registration
        cbManager.connect(peripheral, options: nil)
        cbManager.registerForConnectionEvents(options: nil)
    default:
        // Not interested
    }
}
// Initialization
private var cbManager: CBCentralManager!

cbManager = CBCentralManager(delegate: self, queue: nil)

// Registering for Gatt Connection events
let matchingOptions = [CBConnectionEventMatchingOption.serviceUUIDs : [myServiceUUID]]()

cbManager.registerForConnectionEvents(options: matchingOptions)

// Delegate callback
func centralManager(_ central: CBCentralManager, connectionEventDidOccur event: CBConnectionEvent, for peripheral: CBPeripheral) {
    switch event {
        case .peerConnected:
            // We are interested in this peripheral, clear registration
            cbManager.connect(peripheral, options: nil)
            cbManager.registerForConnectionEvents(options: nil)
        default:
            // Not interested
    }
}
// Initialization
private var cbManager: CBCentralManager!
cbManager = CBCentralManager(delegate: self, queue: nil)

// Registering for Gatt Connection events
let matchingOptions = [CBConnectionEventMatchingOption.serviceUUIDs : [myServiceUUID]]()
cbManager.registerForConnectionEvents(options: matchingOptions)

// Delegate callback
func centralManager(_ central: CBCentralManager, connectionEventDidOccur event: CBConnectionEvent, for peripheral: CBPeripheral) {
    switch event {
    case .peerConnected:
        // We are interested in this peripheral, clear registration
        cbManager.connect(peripheral, options: nil)
        cbManager.registerForConnectionEvents(options: nil)
    default:
        // Not interested
    }
}
private var cbManager: CBCentralManager!
cbManager = CBCentralManager(delegate: self, queue: nil)

let matchingOptions = [CBConnectionEventMatchingOption.serviceUUIDs : [myServiceUUID]]()
cbManager.registerForConnectionEvents(options: matchingOptions)

func centralManager(_ central: CBCentralManager, connectionEventDidOccur event: CBConnectionEvent, for peripheral: CBPeripheral) {
    switch event {
    case .peerConnected:
        // We are interested in this peripheral, clear registration
        cbManager.connect(peripheral, options: nil)
        cbManager.registerForConnectionEvents(options: nil)
    default:
        // Not interested
    }
}
// Initialization
private var cbManager: CBCentralManager!

cbManager = CBCentralManager(delegate: self, queue: nil)

// Registering for Gatt Connection events
let matchingOptions = [CBConnectionEventMatchingOption.serviceUUIDs : [myServiceUUID]]() 

cbManager.registerForConnectionEvents(options: matchingOptions)

// Delegate callback
func centralManager(_ central: CBCentralManager, connectionEventDidOccur event: CBConnectionEvent, for peripheral: CBPeripheral) {
    switch event {
    case .peerConnected:
        // We are interested in this peripheral, clear registration
        cbManager.connect(peripheral, options: nil)
        cbManager.registerForConnectionEvents(options: nil)
    default:
        // Not interested
    }
}
Outgoing Connection
Connecting Out
private var central: CBCentralManager?
central = CBCentralManager(delegate: self, queue: nil)
private var central: CBCentralManager?
central = CBCentralManager(delegate: self, queue: nil)

central?.connect(myPeripheral, options: nil)
Connected
optional func centralManager(_ central: CBCentralManager, didConnect peripheral: CBPeripheral) {
    // Handle connection
}

BR/EDR Connected and Paired
Core Bluetooth Dual-Mode
Improving Dual-Mode Pairing
Improving Dual-Mode Pairing
Cross Transport Key Derivation
Cross Transport Key Derivation

Bluetooth 4.2 SIG Specification
Cross Transport Key Derivation

Bluetooth 4.2 SIG Specification

Single pairing process
Cross Transport Key Derivation

Bluetooth 4.2 SIG Specification

Single pairing process

Same CBPeripheral identifier
Cross Transport Key Derivation

Bluetooth 4.2 SIG Specification
Single pairing process
Same CBPeripheral identifier
Transparent to the application
Cross Transport Key Derivation

Bluetooth 4.2 SIG Specification
Single pairing process
Same CBPeripheral identifier
Transparent to the application
Cross Transport Key Derivation

NEW
Cross Transport Key Derivation

NEW
Cross Transport Key Derivation

![Bluetooth Settings Screen]

- **Bluetooth**: On
- **MY DEVICES**: MyDevice - Connected

*To pair an Apple Watch with your iPhone, go to the Apple Watch app.*
Instead of Inquiry

[Diagram showing the process of an Inquiry Scan and Inquiry Response]
Low Energy Scan

Scan

Advertise
CTKD — Pairing

Bluetooth Pairing Request
“MyDevice” would like to pair with your iPhone. Confirm that this code is displayed on “MyDevice”.

846031

Cancel  Pair

LE Connected

Headphones
Key Derivation

LE Connected and Paired
Key Derivation

LE Connected and Paired
Key Derivation

LE Connected and Paired
CTKD — BR/EDR Connected
CTKD — BR/EDR Connected
Improving Dual-Mode Connections
Improving Dual-Mode Connections

LE Connected and Paired
Improving Dual-Mode Connections

LE Connected and Paired

BR/EDR Connected and Paired
Bridging

NEW
Bridging

Low Energy proximity triggers BR/EDR connection
Bridging

Low Energy proximity triggers BR/EDR connection

Works on devices supporting CTKD
Bridging

Low Energy proximity triggers BR/EDR connection

Works on devices supporting CTKD

```swift
public let CBConnectPeripheralOptionEnableTransportBridgingKey: String
```
Bridging

Low Energy proximity triggers BR/EDR connection

Works on devices supporting CTKD

```swift
public let CBConnectPeripheralOptionEnableTransportBridgingKey: String

cbCentralManager.connect(cbPeripheral,
    options: [CBConnectPeripheralOptionEnableTransportBridgingKey : true])
```
Bridging
Bridging

cbCentralManager.connect(cbPeripheral, options: [CBConnectPeripheralOptionEnableTransportBridgingKey : true])

LE Connected and Paired
Bridging

```swift
cbCentralManager.connect(cbPeripheral, options: 
[CBConnectPeripheralOptionEnableTransportBridgingKey : true])
```
Bridging

BR/EDR Connected and Paired

LE Connected and Paired
Privacy Update

Meghna Lav, Bluetooth Engineer
Privacy matters
Enhancements
Enhancements

User authorization
Enhancements

User authorization

Accessory notifications
User Authorization
How it currently works

Required only for background advertising
User Authorization
User Authorization

User consent required for all Core Bluetooth API’s

“CBClasic” Would Like to Use Bluetooth
We use Bluetooth to discover, connect to, and share information with nearby devices

Don’t Allow | OK
User Authorization

User consent required for all Core Bluetooth API’s

Applies to apps linked with any SDK

“CBClassic” Would Like to Use Bluetooth
We use Bluetooth to discover, connect to, and share information with nearby devices

Don’t Allow  OK
User Authorization

User consent required for all Core Bluetooth API’s

Applies to apps linked with any SDK

Can be modified in Settings

“CBClassic” Would Like to Use Bluetooth
We use Bluetooth to discover, connect to, and share information with nearby devices

Don't Allow  OK
User Authorization

User consent required for all Core Bluetooth API’s

Applies to apps linked with any SDK

Can be modified in Settings

Required on iOS, watchOS, and tvOS
User Authorization on watchOS

"CBSample" would like to use Bluetooth
Use Bluetooth to discover devices near you.

OK

Don't Allow
User Authorization on watchOS

Shared between iOS and watchOS
User Authorization on watchOS

Shared between iOS and watchOS

Except for standalone watchOS applications
<table>
<thead>
<tr>
<th>Key</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bundle identifier</td>
<td>String</td>
<td>$(PRODUCT.Bundle_IDENTIFIER)</td>
</tr>
<tr>
<td>InfoDictionary version</td>
<td>String</td>
<td>6.0</td>
</tr>
<tr>
<td>Bundle name</td>
<td>String</td>
<td>$(PRODUCT_NAME)</td>
</tr>
<tr>
<td>Bundle OS Type code</td>
<td>String</td>
<td>APPL</td>
</tr>
<tr>
<td>Bundle versions string, short</td>
<td>String</td>
<td>1.0</td>
</tr>
<tr>
<td>Bundle version</td>
<td>String</td>
<td>1</td>
</tr>
<tr>
<td>Application requires iPhone environment</td>
<td>Boolean</td>
<td>YES</td>
</tr>
<tr>
<td>Privacy - Bluetooth Always Usage Description</td>
<td>String</td>
<td>Use Bluetooth to discover, connect to, and share information with nearby devices</td>
</tr>
<tr>
<td>Required background modes</td>
<td>Array</td>
<td>(2 items)</td>
</tr>
<tr>
<td>Launch screen interface file base name</td>
<td>String</td>
<td>LaunchScreen</td>
</tr>
<tr>
<td>Main storyboard file base name</td>
<td>String</td>
<td>Main</td>
</tr>
<tr>
<td>Required device capabilities</td>
<td>Array</td>
<td>(1 item)</td>
</tr>
<tr>
<td>Supported interface orientations</td>
<td>Array</td>
<td>(3 items)</td>
</tr>
<tr>
<td>Supported interface orientations (iPad)</td>
<td>Array</td>
<td>(4 items)</td>
</tr>
</tbody>
</table>
Core BluetoothClassicSample: [access] This app has crashed because it attempted to access privacy-sensitive data without a usage description. The app’s Info.plist must contain an NSBluetoothAlwaysUsageDescription key with a string value explaining to the user how the app uses this data.
var authorization: CBManagerAuthorization { get }

enum CBManagerAuthorization : Int {
    init?(.rawValue: Int)
    var rawValue: Int { get }
    case notDetermined
    case restricted
    case denied
    case allowedAlways
}

Adoption
New property
Adoption

New property

```swift
var authorization: CBManagerAuthorization { get }

enum CBManagerAuthorization : Int {
    init?(rawValue: Int)
    var rawValue: Int { get }
    case notDetermined
    case restricted
    case denied
    case allowedAlways
}
```
Adoption
New property

```swift
var authorization: CBManagerAuthorization { get }

enum CBManagerAuthorization : Int {
    init?(rawValue: Int)
    var rawValue: Int { get }
    case notDetermined
    case restricted
    case denied
    case allowedAlways
}
```
func centralManagerDidUpdateState(_ central: CBCentralManager)
func peripheralManagerDidUpdateState(_ peripheral: CBPeripheralManager)

open var state: CBManagerState { get }

// if (state == CBManagerState.unauthorized)
open var authorization: CBManagerAuthorization { get }
func centralManagerDidUpdateState(_ central: CBCentralManager)
func peripheralManagerDidUpdateState(_ peripheral: CBPeripheralManager)

open var state: CBManagerState { get }

// if (state == CBManagerState.unauthorized)
open var authorization: CBManagerAuthorization { get }
func centralManagerDidUpdateState(_ central: CBCentralManager)
func peripheralManagerDidUpdateState(_ peripheral: CBPeripheralManager)

open var state: CBManagerState { get }

// if (state == CBManagerState.unauthorized)
open var authorization: CBManagerAuthorization { get }
func.centralManagerDidUpdateState(_ central: CBCentralManager)
func.peripheralManagerDidUpdateState(_ peripheral: CBPeripheralManager)

open var state: CBManagerState { get }

// if (state == CBManagerState.unauthorized)
open var authorization: CBManagerAuthorization { get }
// Old managerDidUpdateState
func centralManagerDidUpdateState(_ central: CBCentralManager) {
    if (cbState == CBManagerState.poweredOn) {
        // Kick-off bluetooth functionality
    }
}
// Updated managerDidUpdateState

func centralManagerDidUpdateState(_ central: CBCentralManager) {
    switch central.state {
    case .unknown:
        // Handle state
    case .resetting:
        // Handle state
    case .unsupported:
        // Handle state
    case .unauthorized:
        if (central.authorization != CBManagerAuthorization.allowedAlways) {
            // Prompt user to give permission
        }
    case .poweredOn:
        // Handle state
    case .poweredOff:
        // Handle state
    }
}
Enhancements

User authorization

Accessory notifications
Accessory Notifications
Accessory Notifications

Apple Notification Center Service
Accessory Notifications

Apple Notification Center Service

GATT server service
Accessory Notifications

Apple Notification Center Service

GATT server service

Allows accessories to receive notifications from iOS
ANCS Privacy Update

User permission required to share notifications
ANCS Privacy Update

User permission required to share notifications

Prompted when accessory registers for notifications
ANCS Privacy Update

User permission required to share notifications

Prompted when accessory registers for notifications

Permissions can be changed in Settings
ANCS Privacy Update

User permission required to share notifications

Prompted when accessory registers for notifications

Permissions can be changed in Settings
User permission required to share notifications
Prompted when accessory registers for notifications
Permissions can be changed in Settings
New ANCS Privacy API

```swift
public let CBConnectPeripheralOptionRequiresANCS: String

optional func centralManager(_ central: CBCentralManager, didUpdateANCSAuthorizationFor peripheral: CBPeripheral)

open var ancsAuthorized: Bool { get }
```
New ANCS Privacy API

public let CBConnectPeripheralOptionRequiresANCS: String

optional func centralManager(_ central: CBCentralManager, didUpdateANCSAuthorizationFor peripheral: CBPeripheral)

open var ancsAuthorized: Bool { get }
New ANCS Privacy API

```swift
public let CBConnectPeripheralOptionRequiresANCS: String

optional func centralManager(_ central: CBCentralManager, didUpdateANCSAuthorizationFor peripheral: CBPeripheral)

open var ancsAuthorized: Bool { get }
```
New ANCS Privacy API

```swift
public let CBConnectPeripheralOptionRequiresANCS: String

optional func centralManager(_ central: CBCentralManager, didUpdateANCSAuthorizationFor peripheral: CBPeripheral)

open var ancsAuthorized: Bool { get }
```
Best Practices
Invoke Core Bluetooth APIs only when required
Best Practices

Invoke Core Bluetooth APIs only when required

Scan and advertise for a limited duration
Best Practices

Invoke Core Bluetooth APIs only when required

Scan and advertise for a limited duration

Scan for specific service UUID(s)
Best Practices

Invoke Core Bluetooth APIs only when required
Scan and advertise for a limited duration
Scan for specific service UUID(s)
Be transparent
Core Bluetooth PacketLogger

Duy Phan, Bluetooth Engineer
Overview
Overview

Bluetooth packet analysis application
Overview

Bluetooth packet analysis application

Visualizer for packet logs inside sysdiagnose
Overview

Bluetooth packet analysis application

Visualizer for packet logs inside sysdiagnose

Decode all protocols defined by Bluetooth SIG and Apple
Overview

Bluetooth packet analysis application
Visualizer for packet logs inside sysdiagnose
Decode all protocols defined by Bluetooth SIG and Apple
Rich filtering options
Overview

Bluetooth packet analysis application

Visualizer for packet logs inside sysdiagnose

Decode all protocols defined by Bluetooth SIG and Apple

Rich filtering options

Search by text or regex
Overview

Bluetooth packet analysis application

Visualizer for packet logs inside sysdiagnose

Decode all protocols defined by Bluetooth SIG and Apple

Rich filtering options

Search by text or regex

Comment and flag packets
Overview

Bluetooth packet analysis application
Visualizer for packet logs inside sysdiagnose
Decode all protocols defined by Bluetooth SIG and Apple
Rich filtering options
Search by text or regex
Comment and flag packets
Export raw data for analysis
Top Level View
<table>
<thead>
<tr>
<th>Time</th>
<th>Type</th>
<th>Handle</th>
<th>Addr</th>
<th>Decoded Packet</th>
</tr>
</thead>
<tbody>
<tr>
<td>May 08 23:04:39.824</td>
<td>ATT Send</td>
<td>0x0044</td>
<td>00:09:A7:24:26:38</td>
<td>▶ Read By Type Request - Start Handle: 0x0016 - End Handle: 0x0017 - UUID: GATT CI</td>
</tr>
<tr>
<td>May 08 23:04:39.854</td>
<td>ATT Send</td>
<td>0x0044</td>
<td>00:09:A7:24:26:38</td>
<td>▶ Find Information Request - Handle: 0x0009 - Service Changed</td>
</tr>
<tr>
<td>May 08 23:04:39.854</td>
<td>ATT Send</td>
<td>0x0044</td>
<td>00:09:A7:24:26:38</td>
<td>▶ Find Information Response</td>
</tr>
<tr>
<td>May 08 23:04:39.883</td>
<td>ATT Send</td>
<td>0x0044</td>
<td>00:09:A7:24:26:38</td>
<td>▶ Error Response - Attribute Handle: 0x0016 - Error Code: Attribute Not Found (0x</td>
</tr>
<tr>
<td>May 08 23:04:39.884</td>
<td>ATT Send</td>
<td>0x0044</td>
<td>00:09:A7:24:26:38</td>
<td>▶ Read By Type Request - Start Handle: 0x0018 - End Handle: 0xFFFF - UUID: GATT CI</td>
</tr>
<tr>
<td>May 08 23:04:39.897</td>
<td>HCI Event</td>
<td>00:DB:70:00:75:5A</td>
<td>▶ LE - Advertising Report - 1 ReportNormal - Public - 00:DB:70:00:75:5A - -67 dBm</td>
<td></td>
</tr>
<tr>
<td>May 08 23:04:39.897</td>
<td>HCI Event</td>
<td>00:DB:70:00:75:5A</td>
<td>▶ LE - Advertising Report - 1 ReportNormal - Public - 00:DB:70:00:75:5A - -67 dBm</td>
<td></td>
</tr>
<tr>
<td>May 08 23:04:39.914</td>
<td>ATT Receive</td>
<td>0x0044</td>
<td>00:09:A7:24:26:38</td>
<td>▶ Write Request - Handle: 0x0009 - Service Changed - Configuration - Indication</td>
</tr>
<tr>
<td>May 08 23:04:39.914</td>
<td>ATT Send</td>
<td>0x0044</td>
<td>00:09:A7:24:26:38</td>
<td>▶ Write Response</td>
</tr>
<tr>
<td>May 08 23:04:39.929</td>
<td>HCI Event</td>
<td>00:03:4B:3B:7E:AB</td>
<td>▶ LE - Advertising Report - 1 ReportNormal - Public - 00:03:4B:3B:7E:AB - -87 dBm</td>
<td></td>
</tr>
<tr>
<td>May 08 23:04:39.929</td>
<td>HCI Event</td>
<td>00:03:4B:3B:7E:AB</td>
<td>▶ LE - Advertising Report - 1 ReportNormal - Public - 00:03:4B:3B:7E:AB - -87 dBm</td>
<td></td>
</tr>
<tr>
<td>May 08 23:04:39.945</td>
<td>ATT Receive</td>
<td>0x0044</td>
<td>00:09:A7:24:26:38</td>
<td>▶ Read By Type Response - Packets: 4</td>
</tr>
<tr>
<td>May 08 23:04:39.947</td>
<td>ATT Send</td>
<td>0x0044</td>
<td>00:09:A7:24:26:38</td>
<td>▶ Read By Type Request - Start Handle: 0x0023 - End Handle: 0xFFFF - UUID: GATT CI</td>
</tr>
<tr>
<td>Time</td>
<td>Type</td>
<td>Handle</td>
<td>Addr</td>
<td>Decoded Packet</td>
</tr>
<tr>
<td>------------</td>
<td>-------------------</td>
<td>--------</td>
<td>------------------</td>
<td>---------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>May 08 23:04:39.824</td>
<td>ATT Send</td>
<td>0x0044</td>
<td>00:09:A7:24:26:38</td>
<td>▶️ Read By Type Request - Start Handle: 0x0016 - End Handle: 0x0017 - UUID: GATT Client Service Changed</td>
</tr>
<tr>
<td>May 08 23:04:39.854</td>
<td>ATT Receive</td>
<td>0x0044</td>
<td>00:09:A7:24:26:38</td>
<td>▶️ Find Information Request - Handle: 0x0009 - Service Changed</td>
</tr>
<tr>
<td>May 08 23:04:39.854</td>
<td>ATT Send</td>
<td>0x0044</td>
<td>00:09:A7:24:26:38</td>
<td>▶️ Find Information Response</td>
</tr>
<tr>
<td>May 08 23:04:39.883</td>
<td>ATT Receive</td>
<td>0x0044</td>
<td>00:09:A7:24:26:38</td>
<td>▶️ Error Response - Attribute Handle: 0x0016 - Error Code: Attribute Not Found (0xFF)</td>
</tr>
<tr>
<td>May 08 23:04:39.884</td>
<td>ATT Send</td>
<td>0x0044</td>
<td>00:09:A7:24:26:38</td>
<td>▶️ Read By Type Request - Start Handle: 0x0016 - End Handle: 0xFFFF - UUID: GATT Client Service Changed</td>
</tr>
<tr>
<td>May 08 23:04:39.897</td>
<td>HCI Event</td>
<td></td>
<td>00:DB:70:00:75:5A</td>
<td>▶️ LE - Advertising Report - 1 ReportNormal - Public - 00:DB:70:00:75:5A -67 dBm</td>
</tr>
<tr>
<td>May 08 23:04:39.897</td>
<td>HCI Event</td>
<td></td>
<td>00:DB:70:00:75:5A</td>
<td>▶️ LE - Advertising Report - 1 ReportNormal - Public - 00:DB:70:00:75:5A -67 dBm</td>
</tr>
<tr>
<td>May 08 23:04:39.914</td>
<td>ATT Receive</td>
<td>0x0044</td>
<td>00:09:A7:24:26:38</td>
<td>▶️ Write Request - Handle: 0x0009 - Service Changed - Configuration - Indication</td>
</tr>
<tr>
<td>May 08 23:04:39.914</td>
<td>ATT Send</td>
<td>0x0044</td>
<td>00:09:A7:24:26:38</td>
<td>▶️ Write Response</td>
</tr>
<tr>
<td>May 08 23:04:39.929</td>
<td>HCI Event</td>
<td></td>
<td>00:03:4B:3B:7E:AB</td>
<td>▶️ LE - Advertising Report - 1 ReportNormal - Public - 00:03:4B:3B:7E:AB -87 dBm</td>
</tr>
<tr>
<td>May 08 23:04:39.929</td>
<td>HCI Event</td>
<td></td>
<td>00:03:4B:3B:7E:AB</td>
<td>▶️ LE - Advertising Report - 1 ReportNormal - Public - 00:03:4B:3B:7E:AB -87 dBm</td>
</tr>
<tr>
<td>May 08 23:04:39.945</td>
<td>ATT Receive</td>
<td>0x0044</td>
<td>00:09:A7:24:26:38</td>
<td>▶️ Read By Type Response - Packets: 4</td>
</tr>
<tr>
<td>May 08 23:04:39.947</td>
<td>ATT Send</td>
<td>0x0044</td>
<td>00:09:A7:24:26:38</td>
<td>▶️ Read By Type Request - Start Handle: 0x0023 - End Handle: 0xFFFF - UUID: GATT Client Service Changed</td>
</tr>
<tr>
<td>Time</td>
<td>Type</td>
<td>Handle</td>
<td>Addr</td>
<td>Decoded Packet</td>
</tr>
<tr>
<td>------------</td>
<td>--------------</td>
<td>--------</td>
<td>------------</td>
<td>--------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>May 08 23:04:39.824</td>
<td>ATT Send 0x0044</td>
<td>00:09:A7:24:26:38</td>
<td>00:09:A7:24:26:38</td>
<td>▶ Read By Type Request - Start Handle: 0x0016 - End Handle: 0x0017 - UUID: GATT CI</td>
</tr>
<tr>
<td>May 08 23:04:39.854</td>
<td>ATT Receive 0x0044</td>
<td>00:09:A7:24:26:38</td>
<td>00:09:A7:24:26:38</td>
<td>▶ Find Information Request - Handle: 0x0009 - Service Changed</td>
</tr>
<tr>
<td>May 08 23:04:39.854</td>
<td>ATT Send 0x0044</td>
<td>00:09:A7:24:26:38</td>
<td>00:09:A7:24:26:38</td>
<td>▶ Find Information Response</td>
</tr>
<tr>
<td>May 08 23:04:39.884</td>
<td>HCI Event 0x0044</td>
<td>00:09:A7:24:26:38</td>
<td>00:09:A7:24:26:38</td>
<td>▶ Read By Type Request - Start Handle: 0x0018 - End Handle: 0xFFFD - UUID: GATT CI</td>
</tr>
<tr>
<td>May 08 23:04:39.897</td>
<td>HCI Event 0x0044</td>
<td>00:DB:70:00:75:5A</td>
<td>LE - Advertising Report - 1 ReportNormal - Public - 00:DB:70:00:75:5A -67 dBm</td>
<td></td>
</tr>
<tr>
<td>May 08 23:04:39.897</td>
<td>HCI Event 0x0044</td>
<td>00:DB:70:00:75:5A</td>
<td>LE - Advertising Report - 1 ReportNormal - Public - 00:DB:70:00:75:5A -67 dBm</td>
<td></td>
</tr>
<tr>
<td>May 08 23:04:39.914</td>
<td>ATT Receive 0x0044</td>
<td>00:09:A7:24:26:38</td>
<td>00:09:A7:24:26:38</td>
<td>▶ Write Request - Handle: 0x0009 - Service Changed - Configuration - Indication</td>
</tr>
<tr>
<td>May 08 23:04:39.914</td>
<td>ATT Send 0x0044</td>
<td>00:09:A7:24:26:38</td>
<td>00:09:A7:24:26:38</td>
<td>▶ Write Response</td>
</tr>
<tr>
<td>May 08 23:04:39.929</td>
<td>HCI Event 0x0044</td>
<td>00:03:48:3B:7E:AB</td>
<td>LE - Advertising Report - 1 ReportNormal - Public - 00:03:48:3B:7E:AB -87 dBm</td>
<td></td>
</tr>
<tr>
<td>May 08 23:04:39.929</td>
<td>HCI Event 0x0044</td>
<td>00:03:48:3B:7E:AB</td>
<td>LE - Advertising Report - 1 ReportNormal - Public - 00:03:48:3B:7E:AB -87 dBm</td>
<td></td>
</tr>
<tr>
<td>May 08 23:04:39.933</td>
<td>HCI Event 0x0044</td>
<td>00:03:48:3B:7E:AB</td>
<td>LE - Advertising Report - 1 ReportNormal - Public - 00:03:48:3B:7E:AB -87 dBm</td>
<td></td>
</tr>
<tr>
<td>May 08 23:04:39.933</td>
<td>HCI Event 0x0044</td>
<td>00:03:48:3B:7E:AB</td>
<td>LE - Advertising Report - 1 ReportNormal - Public - 00:03:48:3B:7E:AB -87 dBm</td>
<td></td>
</tr>
<tr>
<td>May 08 23:04:39.945</td>
<td>ATT Receive 0x0044</td>
<td>00:09:A7:24:26:38</td>
<td>00:09:A7:24:26:38</td>
<td>▶ Read By Type Response - Packets: 4</td>
</tr>
<tr>
<td>May 08 23:04:39.947</td>
<td>ATT Send 0x0044</td>
<td>00:09:A7:24:26:38</td>
<td>00:09:A7:24:26:38</td>
<td>▶ Read By Type Request - Start Handle: 0x0023 - End Handle: 0xFFFD - UUID: GATT CI</td>
</tr>
</tbody>
</table>
### Hierarchical View

<table>
<thead>
<tr>
<th>Time</th>
<th>Type</th>
<th>Handle</th>
<th>Addr</th>
<th>Decoded Packet</th>
</tr>
</thead>
<tbody>
<tr>
<td>May 08 23:04:39.884</td>
<td>ATT Send</td>
<td>0x0044</td>
<td>00:09:A7:24:26:38</td>
<td>Read By Type Request - Start Handle: 0x0018 - End Handle: 0xFFFF - UUID: GAT...</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Opcode: 0x0008</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Starting Handle: 0x0018</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Ending Handle: 0xFFFF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Attribute Type: 2803 (GATT Characteristic Declaration)</td>
</tr>
<tr>
<td>May 08 23:04:39.884</td>
<td>L2CAP Send</td>
<td>0x0044</td>
<td>00:09:A7:24:26:38</td>
<td>Channel ID: 0x0004 Length: 0x0007 (07) [ 08 18 00 FF FF 03 28 ]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>L2CAP Payload: 00000000 0818 00FF FF03 28</td>
</tr>
<tr>
<td>May 08 23:04:39.884</td>
<td>ACL Send</td>
<td>0x0044</td>
<td>00:09:A7:24:26:38</td>
<td>Data [Handle: 0x0044, Packet Boundary Flags: 0x0, Length: 0x000B (11)]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Broadcast Flags: [00] 0x00 - Reserved For Future Use</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Data (0x000B Bytes)</td>
</tr>
<tr>
<td>May 08 23:04:39.884</td>
<td>ACL Send</td>
<td>0x0044</td>
<td>00:09:A7:24:26:38</td>
<td>00000000 4400 0B00 0700 0400 0818 00FF FF03 28 D...</td>
</tr>
<tr>
<td>May 08 23:04:39.945</td>
<td>ATT Receive</td>
<td>0x0044</td>
<td>00:09:A7:24:26:38</td>
<td>Read By Type Response - Packets: 4</td>
</tr>
<tr>
<td>May 08 23:04:39.947</td>
<td>ATT Send</td>
<td>0x0044</td>
<td>00:09:A7:24:26:38</td>
<td>Read By Type Request - Start Handle: 0x0023 - End Handle: 0xFFFF - UUID: GAT...</td>
</tr>
</tbody>
</table>
### Hierarchical View

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Event</th>
<th>Type</th>
<th>Value</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>May 08</td>
<td>23:04:39.884</td>
<td>ATT Send</td>
<td>0x0044</td>
<td>00:09:A7:24:26:38</td>
<td>Read By Type Request - Start Handle: 0x0018 - End Handle: 0xFFFF - UUID: GATT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Opcode: 0x0008</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Starting Handle: 0x0018</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Ending Handle: 0xFFFF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Attribute Type: 2803 (GATT Characteristic Declaration)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Event</th>
<th>Type</th>
<th>Value</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>May 08</td>
<td>23:04:39.884</td>
<td>L2CAP Send</td>
<td>0x0044</td>
<td>00:09:A7:24:26:38</td>
<td>Channel ID: 0x0004 Length: 0x0007 (07) [ 08 18 00 FF FF 03 28 ]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Channel ID: 0x0004 Length: 0x0007 (07) [ 08 18 00 FF FF 03 28 ]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>L2CAP Payload:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>00000000: 0818 00FF FF03 28</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>.........(</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Event</th>
<th>Type</th>
<th>Value</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>May 08</td>
<td>23:04:39.884</td>
<td>ACL Send</td>
<td>0x0044</td>
<td>00:09:A7:24:26:38</td>
<td>Data [Handle: 0x0044, Packet Boundary Flags: 0x00, Length: 0x0000B (11)]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Packet Boundary Flags: [00] 0x00 - Reserved For Future Use</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Broadcast Flags: [00] 0x00 - Point-to-point</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Data (0x000B Bytes)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>00000000: 4400 0B00 0700 0400 0818 00FF FF03 28</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>00000000: 4400 0B00 0700 0400 0818 00FF FF03 28</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Event</th>
<th>Type</th>
<th>Value</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>May 08</td>
<td>23:04:39.884</td>
<td>ACL Send</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>Time</td>
<td>Event</td>
<td>Data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>----------------------</td>
<td>-------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>May 08</td>
<td>23:04:39.884</td>
<td>ATT Send</td>
<td>0x0044</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Read By Type Request</strong> - <strong>Start Handle</strong>: 0x0018 - <strong>End Handle</strong>: 0xFFFF - <strong>UUID</strong>: GAT</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Opcode</strong>: 0x0008</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Starting Handle</strong>: 0x0018</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Ending Handle</strong>: 0xFFFF</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Attribute Type</strong>: 2803 (GATT Characteristic Declaration)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>May 08</td>
<td>23:04:39.884</td>
<td>L2CAP Send</td>
<td>0x0044</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Channel ID</strong>: 0x0004 - <strong>Length</strong>: 0x0007 (07) [ 08 18 00 FF FF 03 28 ]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Channel ID</strong>: 0x0004 - <strong>Length</strong>: 0x0007 (07) [ 08 18 00 FF FF 03 28 ]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>L2CAP Payload</strong>:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>00000000: 0818 00FF FF03 28</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>May 08</td>
<td>23:04:39.884</td>
<td>ACL Send</td>
<td>0x0044</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Data</strong> [<strong>Handle</strong>: 0x0044, <strong>Packet Boundary Flags</strong>: 0x0, <strong>Length</strong>: 0x000B (11)]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Packet Boundary Flags</strong>: [00] 0x00 - Reserved For Future Use</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Broadcast Flags</strong>: [00] 0x00 - Point-to-point</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Data</strong> [0x000B Bytes]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>May 08</td>
<td>23:04:39.884</td>
<td>ACL Send</td>
<td>0x0044</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Data</strong> [4400 0800 0700 0400 0818 00FF FF03 28]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Data</strong> [4400 0B00 0700 0400 0818 00FF FF03 28]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Live Capture
Live Capture
Live Capture
Install iOS 13 developer beta
Live Capture

Install iOS 13 developer beta

Install iOS Bluetooth developer logging profile
Live Capture

Install iOS 13 developer beta

Install iOS Bluetooth developer logging profile

Launch PacketLogger
Live Capture

Install iOS 13 developer beta

Install iOS Bluetooth developer logging profile

Launch PacketLogger

Connect your iOS device to your Mac
Live Capture

Install iOS 13 developer beta

Install iOS Bluetooth developer logging profile

Launch PacketLogger

Connect your iOS device to your Mac

Select File and "New iOS Trace"
Live Capture

Install iOS 13 developer beta
Install iOS Bluetooth developer logging profile
Launch PacketLogger
Connect your iOS device to your Mac
Select File and “New iOS Trace”
Indicator will appear on iOS device
Getting PacketLogger
Getting PacketLogger

Download “Additional Tools for Xcode”
Getting PacketLogger

Download “Additional Tools for Xcode”

PacketLogger is inside the Hardware folder
Getting PacketLogger

Download “Additional Tools for Xcode”

PacketLogger is inside the Hardware folder

For best performance run with macOS Catalina
Summary
Summary

Use chipset with latest Bluetooth standard
Summary

Use chipset with latest Bluetooth standard

Build Core Bluetooth apps for BR/EDR devices
Summary

Use chipset with latest Bluetooth standard

Build Core Bluetooth apps for BR/EDR devices

Protect user privacy
Summary

Use chipset with latest Bluetooth standard

Build Core Bluetooth apps for BR/EDR devices

Protect user privacy

Take advantage of the developer beta
Summary

Use chipset with latest Bluetooth standard

Build Core Bluetooth apps for BR/EDR devices

Protect user privacy

Take advantage of the developer beta

Refer to Accessory Design Guidelines for Apple devices
Summary

Use chipset with latest Bluetooth standard

Build Core Bluetooth apps for BR/EDR devices

Protect user privacy

Take advantage of the developer beta

Refer to Accessory Design Guidelines for Apple devices

Apple is here to help
More Information

developer.apple.com/wwdc19/901

Core Bluetooth Lab

Friday, 4:00