

Core Bluetooth

プログラミングガイド

注意: アプリケーションの用途によって、試みるべき方法とその順序は違うかも知れません。第1の方法はまったく試みない、第1と第2の方法を並行して試みる、などの流れも考えられます。

既知のペリフェラルのリストを取得する

システムは初めて検出したペリフェラルに対して、識別子（NSUUIDオブジェクトで表されるUUID）を生成します。この識別子を（たとえばNSUserDefaultsクラスのリソースを使って）保存しておけば、後でこれを使って、CBCentralManagerクラスのretrievePeripheralsWithIdentifiers:メソッドで再接続できます。以下、このメソッドを使って、一度接続したことがあるペリフェラルに再接続する方法を示します。

アプリケーションの起動時に、これまでに検出、接続し（、その識別子を保存しておい）たペリフェラルの識別子をまとめた配列を渡して、次のようにretrievePeripheralsWithIdentifiers:メソッドを実行します。

```
knownPeripherals =  
    [myCentralManager retrievePeripheralsWithIdentifiers:savedIdentifiers];
```

セントラルマネージャは、指定された識別子を、過去に検出したペリフェラルの識別子と照合し、その結果をCBPeripheralオブジェクトの配列として返します。ひとつも合致しなければ配列は空なので、ほかの方法を試みることになります。そうでなければ、（適当なUIを設けて）再接続先をユーザが選択できるようにしてください。

ユーザがペリフェラルを選択したら、CBCentralManagerクラスのconnectPeripheral:options:メソッドで接続を試みます。接続が可能であれば、セントラルマネージャはデリゲートオブジェクトのcentralManager:didConnectPeripheral:メソッドを呼び出します。この時点でペリフェラルは、再接続された状態になっています。

注意: 何らかの理由で、ペリフェラルに接続できないこともあります。たとえば、セントラルから離れたところに移動してしまっているかも知れません。さらに、BLE機器によっては、アドレスが定期的に、ランダムに変わります。すぐ近くにあったとしても、システムが検出した時点以降にアドレスが変わっていれば、接続しようとしているCBPeripheralオブジェクトと実際のペリフェラルが対応しないことになります。これが原因で再接続できなかった場合、改めてscanForPeripheralsWithServices:options:メソッドで検出しなければなりません。

ランダムに変わるアドレスについては、Bluetooth 4.0仕様のVolume 3、Part C、Section 10.8および「[Bluetooth Accessory Design Guidelines for Apple Products](#)」を参照してください。

接続済みペリフェラルのリストを取得する

ペリフェラルに再接続するもうひとつの方法として、当該ペリフェラルが（たとえば他のアプリケーションによって）システムに接続済みでないか調べる、というものがあります。これはCBCentralManagerクラスのretrieveConnectedPeripheralsWithServices:メソッドで行います。戻り値は、接続済みペリフェラルを表すCBPeripheralオブジェクトの配列です。

複数のペリフェラルが接続されているかも知れないので、（サービスのUUIDを表す）CBUUIDオブジェクトの配列を引数として渡します。システムに接続済みで、かつ、指定したサービスを提供するペリフェラルのみ取得できます。該当するペリフェラルがなければ配列は空なので、ほかの方法を試みることになります。そうでなければ、（適当なUIを設けて）再接続先をユーザが選択できるようにしてください。

ユーザがペリフェラルを選択したら、CBCentralManagerクラスのconnectPeripheral:options:メソッドで、当該アプリケーションにローカルに接続します（システムには接続済みでも、アプリケーションと接続しなければデータのやり取りはできません）。ローカルな接続が確立すると、セントラルマネージャはデリゲートオブジェクトのcentralManager:didConnectPeripheral:メソッドを呼び出します。この時点でペリフェラルは、再接続された状態になっています。

ローカルデバイスをペリフェラルとしてセットアップするためのベストプラクティス

CoreBluetoothフレームワークは、セントラル側のさまざまなトランザクションだけでなく、ペリフェラルとしての役割を実装するためにも有用です。この章では、これを的確に行うためのガイドラインやベストプラクティスを紹介します。

アドバタイズに関する検討事項

ローカルデバイスにペリフェラルとしての役割を実装する上で、データのアドバタイズ機能は重要な要素です。以下の各節で、適切に実装するための考慮事項を説明します。

アドバタイズするデータの制限事項を考慮する

ペリフェラルのデータをアドバタイズするためには、当該データの辞書を渡して、`CBPeripheralManager`クラスの`startAdvertising:メソッド`を実行します（“[サービスをアドバタイズする](#)”（27ページ）を参照）。辞書を生成する際、何をいくつアドバタイズできるか、に関して制限があることを考慮してください。

アドバタイズパケットには一般に、ペリフェラルに関するさまざまな情報を収容できますが、アドバタイズするのはデバイスのローカル名とサービスのUUIDだけです。したがって、アドバタイズ辞書のキーとして、`CBAdvertisementDataLocalNameKey`と`CBAdvertisementDataServiceUUIDsKey`の2つだけを指定します。他のキーを指定すればエラーになります。

さらに、アドバタイズの容量にも制限があります。アプリケーションがフォアグラウンド状態であれば、2つのアドバタイズデータキーを組み合わせ、当初は28バイト分までのアドバタイズデータを指定できます。これで不足する場合、走査に対する応答で、ローカル名に用いる10バイトを追加できます。この領域に入りきらないサービスUUIDは、特別な「オーバーフロー」領域に追加します。これを検出するためには、明示的に当該UUIDを指定して走査しなければなりません。一方、バックグラウンド状態の場合、ローカル名はアドバタイズの対象外であり、サービスUUIDはすべてオーバーフロー領域に入ります。

注意: この長さには、ヘッダ情報の2バイト（新しいデータ型ごとに必要）を含みません。アドバタイズデータや応答データの具体的な形式は、Bluetooth 4.0仕様のVolume 3、Part C、Section 11に定義されています。

この制限内に収まるよう、主サービスのUUIDのみアドバタイズするようにしてください。

必要のないデータはアドバタイズしない

ペリフェラルのデータをアドバタイズするにはローカルデバイスの電波（および電池）を使うことになるので、他のデバイスに接続してほしいときのみ、アドバタイズを出すようにしてください。いったん接続すれば、アドバタイズパケットがなくても、データを直接調査し、やり取りできます。したがって、電波の使用を最小限に抑え、処理性能を改善し、電池の寿命を延ばすためにも、必要なくなったらアドバタイズを止めてください。これがBLEトランザクション本来の使い方です。アドバタイズを止めるには、`CBPeripheralManager`クラスの`stopAdvertising`メソッドを使います。

```
[myPeripheralManager stopAdvertising];
```

アドバタイズを出すか否かユーザが判断できるようにする

アドバタイズを出すべきか否か、ユーザにしか判断できないことも少なくありません。たとえば、近くに他のBLE機器がないと分かっている場合、アドバタイズしても無意味です。これをアプリケーションが判断するのは難しいので、ユーザが判断できるよう、適当なユーザインターフェイスを設けるとよいでしょう。

特性を設定する

可変の特性に対しては、プロパティ、値、操作権限を設定することになります。これによって、セントラルがどのようにアクセスし、特性値をやり取りするか、が決まります。特性のプロパティや操作権限は、アプリケーションの必要性に応じて設定してください。以下、次の2通りを想定して、いくつかガイダンスを示します。

- セントラルによる特性値の通知申し込みを受け付ける場合
- 重要な特性値について、ペアリングしたセントラルからのアクセス以外を拒否する場合

通知の申し込みに対応できるように特性を設定する

“頻繁に変化する特性については、値の変化を通知するよう申し込む”（43 ページ）で説明したように、頻繁に変化する（リモートペリフェラルのサービスの）特性値については、通知機能を用いるよう推奨します。できるだけ、特性値の変化を通知するよう、セントラルが申し込めるようにしてください。

可変の特性を生成する際、通知の申し込みに対応できるようにするためには、特性のプロパティとして定数 `CBCharacteristicPropertyNotify` を指定します。

```
myCharacteristic = [[CBMutableCharacteristic alloc]
    initWithType:myCharacteristicUUID
    properties:CBCharacteristicPropertyRead | CBCharacteristicPropertyNotify
    value:nil permissions:CBAAttributePermissionsReadable];
```

この例に示した特性値は読み取り可能であり、セントラルからの通知申し込みにも対応できます。

重要なデータについて、ペアリングした機器からのアクセスのみを許可する

用途によっては、特性値のやり取りに際して、セキュリティを考慮しなければならないかも知れません。たとえばソーシャルメディアサービスを考えてみましょう。このサービスでは、会員の名前や電子メールアドレスなど、個人情報特性として扱うかも知れません。そして、電子メールアドレスについては、信頼性を確認したデバイスからのアクセスしか許可しない、という要件も考えられます。

これを実現するためには、特性のプロパティと操作権限を適切に設定する必要があります。そのコード例を示します。

```
emailCharacteristic = [[CBMutableCharacteristic alloc]
    initWithType:emailCharacteristicUUID
    properties:CBCharacteristicPropertyRead
    | CBCharacteristicPropertyNotifyEncryptionRequired
    value:nil permissions:CBAAttributePermissionsReadEncryptionRequired];
```

この例では、信頼できるデバイスにしか読み取りや通知を許可しないよう、特性を設定しました。リモートのセントラルは、接続後、特性値を読み取ろうと（または通知を申し込もうと）します。すると `Core Bluetooth` は、ローカルのペリフェラルとペアリング手続きをして、信頼性を確認しようと試みます。

セントラルとペリフェラルがどちらもiOSデバイスであれば、両方に、この手続きを行う旨の画面が現れます。セントラルの画面に現れているコードを、ペリフェラルのテキストフィールドに正しく入力すれば、手続きは完了です。

ペリフェラルはこのセントラルが信頼できるものと判断し、暗号化を施した特性値へのアクセスを許可します。

書類の改訂履歴

この表は「Core Bluetooth プログラミングガイド」の改訂履歴です。

| 日付 | メモ |
|--------------------|---|
| \$RevisionDate_2.0 | iOS 7およびOS X v10.9に合わせて、バックグラウンドで長時間にわたるアクションを実行する方法や、周辺機器(ペリフェラル)を取得する新しいメソッドについての情報を改訂しました。 |
| \$RevisionDate_1.0 | 新規資料。Core Bluetoothフレームワークを使って、BLE (Bluetooth Low Energy) 技術を活用したアプリケーションを開発する方法の解説。 |



Apple Inc.
Copyright © 2013 Apple Inc.
All rights reserved.

の法的権利を与え、地域によってはその他の権利がお客様に与えられる場合もあります。

本書の一部あるいは全部を Apple Inc. から書面による事前の許諾を得ることなく複製複製（コピー）することを禁じます。また、製品に付属のソフトウェアは同梱のソフトウェア使用許諾契約書に記載の条件のもとでお使いください。書類を個人で使用する場合に限り1台のコンピュータに保管すること、またその書類にアップルの著作権表示が含まれる限り、個人的な利用を目的に書類を複製することを認めます。

Apple ロゴは、米国その他の国で登録された Apple Inc. の商標です。

キーボードから入力可能な Apple ロゴについても、これを Apple Inc. からの書面による事前の許諾なしに商業的な目的で使用すると、連邦および州の商標法および不正競争防止法違反となる場合があります。

本書に記載されているテクノロジーに関しては、明示または黙示を問わず、使用を許諾しません。本書に記載されているテクノロジーに関するすべての知的財産権は、Apple Inc. が保有しています。本書は、Apple ブランドのコンピュータ用のアプリケーション開発に使用を限定します。

本書には正確な情報を記載するように努めました。ただし、誤植や制作上の誤記がないことを保証するものではありません。

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
U.S.A.

アップルジャパン株式会社
〒163-1450 東京都新宿区西新宿
3丁目20番2号
東京オペラシティタワー
<http://www.apple.com/jp/>

Offline copy. Trademarks go here.

Apple Inc. は本書の内容を確認しておりますが、本書に関して、明示的であるか黙示的であるかを問わず、その品質、正確さ、市場性、または特定の目的に対する適合性に関して何らかの保証または表明を行うものではありません。その結果、本書は「現状有姿のまま」提供され、本書の品質または正確さに関連して発生するすべての損害は、購入者であるお客様が負うものとなります。

いかなる場合も、Apple Inc. は、本書の内容に含まれる瑕疵または不正確さによって生じる直接的、間接的、特殊的、偶発的、または結果的損害に対する賠償請求には一切応じません。そのような損害の可能性があらかじめ指摘されている場合においても同様です。

上記の損害に対する保証および救済は、口頭や書面によるか、または明示的や黙示的であるかを問わず、唯一のものであり、その他一切の保証にかわるものです。Apple Inc. の販売店、代理店、または従業員には、この保証に関する規定に何らかの変更、拡張、または追加を加える権限は与えられていません。

一部の国や地域では、黙示あるいは偶発的または結果的損害に対する賠償の免責または制限が認められていないため、上記の制限や免責がお客様に適用されない場合があります。この保証はお客様に特定