iOSツールワークフローガイド

Developer

目次

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

図、表、リスト

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

iOS用ツールのワークフローについて

iOSアプリケーションを開発するには、Appleの優れた統合開発環境（IDE）であるXcodeを使用します。Xcodeは、アプリケーションのユーザインターフェイスの設計と、それを動作させるためのコードの記述に必要なすべてのツールを備えています。

At a Glance

このドキュメントでは、iOSアプリケーションを開発、公開するために必要な作業の進め方を解説します。開発デバイス上でアプリケーションを実行すること、App Storeでの公開前にユーザに配布してテストしてもらうことも含みます。

このドキュメントの内容は、Xcode 4.3およびiOS SDK 5.0を対象としています。

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

重要な開発アセットを入手、管理する

iOSデバイス上でiOSアプリケーションを実行するためには、iOSアプリケーション開発者としての認証に用いるコード署名証明書など、重要な開発アセットを入手する必要があります。

関連する章 “開発アセット、配布アセットの設定”（9 ページ）

プロジェクト例からiOSのコーディング技法を学ぶ

iOSサンプルコードプロジェクトを調べ、分析することにより、個々の技法やAPIの使い方を効果的に学習できます。サンプルプロジェクトのコードを自身のプロジェクトに取り込んで、手間を省くこともできるでしょう。

関連する節 “サンプルアプリケーションの実行”（23 ページ）

アプリケーションにおけるiCloudストレージアクセスの設定をする

iCloudストレージを利用するためには、アプリケーションのiCloudエンタイトルメントを指定する必要があります。

関連する節 “iCloudエンタイトルメントの設定”（17 ページ）

Mac上でiOSの動作をシミュレートする

アプリケーションのユーザインターフェイスを設計する際には、Mac上のiOSシミュレータでその動作をシミュレートするとよいでしょう。アプリケーションが実際のデバイス上ではどのように動作するか、何らかの発想を得ることができるかも知れません。

関連する章 “iOSシミュレータの使用”（33 ページ）

新しいiOSデバイス上でアプリケーションを実行する

開発用のiOSデバイスを入手したら、アプリケーションをインストール、実行できるよう、必要な設定をしてください。

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

関連する節 “開発用デバイスの準備”（10 ページ）

アプリケーションが想定通りに動作するかどうか確認する

アプリケーションに変更を施すと、バグを紛れ込ませてしまう可能性があります。これまでの作業を無にしてしまうことのないよう、開発戦略の中に単体テストを組み入れてください。

関連する章 “高い品質と性能の保証”（37 ページ）

実ユーザによるアプリケーションテストを実施する

ユーザが実際に使うのとまったく同じ状況で、アプリケーションをテストすることはできません。扱うデータも使い方も異なるからです。App Storeにアプリケーションを公開する前に、実際の環境でテストを行い、できるだけ問題を解消してください。

関連する節 “ユーザテスト用アプリケーションの公開”（42 ページ）

Prerequisites

プログラミングの基本概念にも精通している必要があります。また、次のような概念や技術も把握しておかなければなりません。

- *Developing for the App Store*
- *iOS App Programming Guide*

関連項目

iOS SDKについて詳しくは、http://developer.apple.com/devcenter/iosを参照してください。

Xcodeについてはhttp://developer.apple.com/xcodeを参照してください。

注意 XcodeはMac上で動作します。

iOS Webアプリケーションを開発する場合は、http://developer.apple.com/devcenter/safari/libraryを参照してください。

開発アセット、配布アセットの設定

登録アップルデベロッパになると、iOSデベロッパ向け資料にアクセスし、シミュレータ上で動作するiOSアプリケーションを構築できるようになります（登録アップルデベロッパになるには、http://developer.apple.com/jp/programs/registerにアクセスしてください）。ただし、登録アップルデベロッパになってもiOSデバイスでアプリケーションを実行することはできません。それには、開発チームのメンバーになる必要もあります。詳細については"開発チームへの加入" （9 ページ）を参照してください。

この章では、iOS開発のためにMacやiOSデバイスを設定する方法を示します。また、開発中のアプリケーションをデバイスにインストールするために必要なデジタルIDを保護する方法、アプリケーションテスターやApp Storeの顧客に配布する方法についても説明します。

このドキュメントの対象　以下の内容は、Xcode 4.3およびiOS SDK 5.0を対象としています。

開発チームへの加入

開発中のアプリケーションをiOSデバイスにインストールするためには、*開発チーム* に属する必要があります。**開発チーム**とは、法的主体（個人、会社、組織）や個人（チームのアプリケーションの開発や配布に関与する人）が連携し、オンライン開発リソース（iOS/Macプロビジョニングポータル）を利用する場のことです。チームに属すれば、開発デバイス上でアプリケーションを構築、実行し、顧客に配布することができます。アプリケーションを配布する権限があれば、これをテスターに送り、現実的な環境で実行した結果のフィードバックを得ることができます。さらに、App Storeでアプリケーションを公開することも可能です。

開発チームに加入する方法は2つあります。

- 権限ある人からの招待を受ける方法。
- iOSデベロッパプログラムに加入し、独自のチームを作る方法。

  プログラムに加入する手続きについては、http://developer.apple.com/programs/iosを参照してください。

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

開発用デバイスの準備

iOSシミュレータを利用すると、iOSベースのデバイスを使用せずに、iOSアプリケーションの開発を始めることができます。シミュレータを使って、アプリケーション開発に使用されるAPIと開発ワークフローに慣れることができます。ただし、アプリケーションを公開する前に実際のデバイス上でアプリケーションをテストして、アプリケーションが確実に意図した通りに動作するようにし、実際のハードウェア上でパフォーマンスを調整する必要があります。

> **注意** チームにおける任務がアプリケーションの配布だけで、開発デバイスは扱わないのであれば、この節を飛ばして"配布専用アセットの準備" （15 ページ）に進んでください。

この節では、iOSデバイスを開発用に設定する方法を示します。また、実行中のアプリケーションの画面ショットを撮る方法も説明します。

開発用デバイスの設定

開発中のアプリケーションをデバイス上で実行するには、デバイスを開発用に設定する必要があります。次のような作業が必要ですが、いずれもXcode上で実行できます。

1. 開発者証明書の取得。アプリケーションに署名するために必要です。
2. プロビジョニングプロファイルの取得。開発者証明書、デバイス、その上で実行するアプリケーションを特定するために使います。

デバイスでアプリケーションを実行するには、MacとデバイスをiOS開発用に設定する必要があります。この節では、Mac上でiOSアプリケーションを開発し、デバイス上で実行するために必要なものについて、概要を説明します。

> **注意** iOSベースのデバイスを開発用に構成しても、通常の操作には影響ありません。

開発用にデバイスを準備する際には、次のデジタルアセットをXcodeで作成または取得してください。

- **証明書要求（Certificate Request）**。証明書要求（または証明書署名要求（CSR、Certificate Signing Request））には、*開発用証明書*の生成に用いる個人情報が含まれています。
- **開発用証明書（Development Certificate）**。開発用証明書は、アプリケーションのデベロッパである旨を識別します。キーチェーンに格納する場合、秘密鍵も組にして格納することになります。開発チームの署名アセットや*プロビジョニングプロファイル*には公開鍵だけが入ります。

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

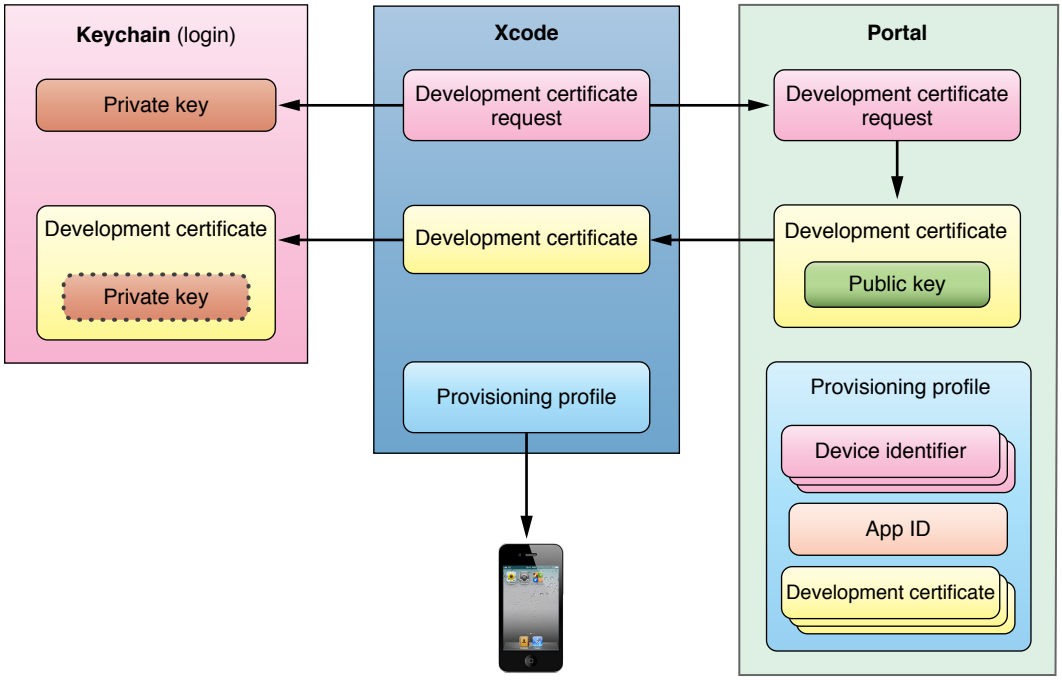
Xcodeは、デバイスにインストールするアプリケーションをビルドする際、キーチェーンの開発者証明書を検索します。証明書が見つかれば、これを使ってアプリケーションに署名するようになっています。見つからなければエラーになります。

- **プロビジョニングプロファイル（Provisioning Profile）。**プロビジョニングプロファイルは、1つ以上の開発用証明書およびデバイスに、アプリケーションIDを関連付けます。

  開発用証明書を使って署名したアプリケーションをデバイスにインストールするためには、デバイスに少なくとも1つのプロビジョニングプロファイルをインストールする必要があります。このプロビジョニングプロファイルは、デベロッパ（開発者証明書によって識別する）とデバイス（デバイス識別子によって識別する）を識別できなければなりません。複数の開発者から成るiOS開発チームに属している場合、チームのメンバは、適切に定義されたプロビジョニングプロファイルがあれば、ほかのメンバがビルドしたアプリケーションを自分のデバイス上で実行できます。

図 1-1に、これらのデジタルアセットの関係を示します。

図 1-1　iOSでの開発に必要なデジタルアセット



Important　デバイスを開発に使えるようにするためには、iOS開発チームに入る必要があります。具体的な手順については、“開発チームへの加入”（9 ページ）を参照してください。

先に進む前に、自分がiOS開発チームにどのような資格（エージェント、管理者、メンバー）で加わっているか、確認しておく必要があります。iOSプロビジョニングポータルにおけるアクセス権限は、次のようにして調べることができます。

1. ポータルにログインしてください。
2. 「Member Center」をクリックします。
3. 「People」をクリックします。

汎用的な開発用にデバイスを設定する

チームの管理者またはエージェントの場合

管理者またはエージェントであれば、汎用的な開発用にデバイスを設定してください。Xcodeは、（必要ならば）開発用証明書をキーチェーンに登録し、デバイスにチームプロビジョニングプロファイルをインストールします。

チームのメンバの場合

汎用的な開発用にデバイスを設定するには、次の手順に従います。

1. 「ウインドウ(Window)」＞「オーガナイザ(Organizer)」を選び、オーガナイザ(Organizer)ウインドウを開きます。
2. 「デバイス(Devices)」をクリックして、デバイスオーガナイザを開きます。
3. デバイスを接続し、デバイスリストからそれを選択します。
4. 「Use for Development」をクリックしてください。
5. 「Identifier」テキストフィールドからデバイス識別子をコピーします。
6. デバイス識別子をチームのエージェントに知らせて、チームのデバイスリストに追加するよう要請してください。

   以下の作業は、追加した旨の連絡を受けてから行います。
7. デバイスオーガナイザで、「Library」セクションの「Provisioning Profiles」を選択し、「Refresh」をクリックします。
8. 開発者証明書がない場合、Xcodeが代わって取得要求を行います。
   a. Xcodeに、開発者証明書を要求するよう指示してください。
   b. チームのエージェントに、Xcodeが開発者証明書を要求した旨を通知してください。

   以下の作業は、エージェントから開発者証明書を発行した旨の連絡を受けた後に行います。
9. デバイスがMacに接続されていること、デバイスオーガナイザに列挙されていることを確認します。
10. 「Library」セクションの「Provisioning Profiles」を選択し、「Refresh」をクリックしてください。

    Xcodeは、（必要ならば）開発用証明書をキーチェーンに登録し、デバイスにチームプロビジョニングプロファイルをインストールします。

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

特殊な開発用にデバイスを設定する

アプリケーション開発に特殊な機能（iCloudストレージ、Push Notification、In-App Purchase、Game Centerなど）を要する場合は、それに応じてデバイスを設定する必要があります。

特殊な開発用にデバイスを設定するには、次の手順に従います。

1. iOSプロビジョニングポータルを調べ、チームのデバイスリストに当該デバイスが載っていない場合は、“汎用的な開発用にデバイスを設定する”（12 ページ）の手順で追加してください。
2. アプリケーションの特殊な要件を記述したプロビジョニングプロファイルが、ポータルにあるかどうか確認します。

   ない場合は、チームのエージェントまたは管理者が作成し、開発者証明書とデバイスに追加してから、以下の作業に進んでください。
3. デバイスオーガナイザで、「Library」セクションの「Provisioning Profiles」を選択し、「Refresh」をクリックします。

   特殊な要件を記述したプロビジョニングプロファイルが、プロファイルリストに載っているはずです。ない場合、複数のiOS開発チームに属しているとすれば、「Refresh」をクリックした後、正しいiOS開発チームの証明書を入力しなかったのが原因かも知れません。
4. デバイスがMacに接続されていることを確認します。
5. 特殊な要件を記述したプロビジョニングプロファイルを、リストからデバイス上にドラッグしてください。

Important　特殊な要件を記述したプロビジョニングプロファイルをデバイスにインストールした後、アプリケーションのビルド対象が、当該プロビジョニングプロファイルを使ってアプリケーションに署名することを確認しておきます。詳細については、“コード署名IDの設定”（26 ページ）を参照してください。

デバイスにiOSをインストール

iOSアプリケーションを開発する際には、その対象であるデバイスとiOSリリースを組み合わせた環境で、実際の動作を確認しなければなりません。

© 2012 Apple Inc. All Rights Reserved.

Important デバイスのOSをアップグレードすると、元に戻す（旧OSをインストールする）ことはできなくなります。

公開されているiOSバージョンのインストール 最新の公開iOSバージョンをダウンロードし、デバイスにインストールするには、iTunesを使用します。

ベータ版iOSバージョンのインストール ベータ版のXcodeやiOSは、iOS Dev Center（http://developer.apple.com/devcenter/ios）からダウンロード可能です。

ベータ版のiOSをダウンロードするには、iOS開発チームのメンバーでなければなりません。

ベータ期間中、iOSのベータ版をインストールするのは、ベータ版専用に用意したデバイスのみにしてください。

ベータ版のXcodeでアプリケーションをApp Storeに登録することはできません。

入手したベータ版iOSをデバイスにインストールしますインストール完了後、デバイスオーガナイザで当該デバイスを選択し、「Use for Development」をクリックしてください。

スクリーンショットのキャプチャ

スクリーンショットは、アプリケーションの仕様をドキュメントにまとめる際に役立つほか、ユーザがアプリケーションのアイコンをタップしたときに、iOSによって表示される、アプリケーションの起動画面を作成するためにも使います。

- **接続されたデバイスから、Xcodeにスクリーンショットを取り込む方法。**

  デバイスオーガナイザで、Macに接続されたデバイス上で動作しているアプリケーションのスクリーンショットを撮ることができます。

  この画像をアプリケーションの起動画面にすることも可能です。

  スクリーンショットのPNGファイルを取得するには、それをデスクトップにドラッグします。

- **デバイス上でスクリーンショットを撮る方法。**

  デバイス上で直接スクリーンショットを取り、iPhotoアプリケーションを使ってMac上に取り込むことも可能です。

  デバイス上でスクリーンショットをキャプチャするには、「ロック(Lock)」ボタンと「ホーム(Home)」ボタンを同時に押します。スクリーンショットが「写真(Photos)」アプリケーションの「カメラロール(Saved Photos)」アルバムに保存されます。

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

注意 スクリーンショットをキャプチャすると、起動画面には、表示されている通りのステータスバーが含まれますが、iOSは、アプリケーションの起動時にそれを現在のステータスバーに置き換えます。

配布専用アセットの準備

アプリケーションを配布するためには、配布用証明書と配布用プロビジョニングプロファイルが必要です。

- 配布用証明書。配布用証明書は開発チームを識別するために用います。キーチェーンに格納する場合、チームの秘密鍵も格納することになります。チームの署名アセットや配布用プロビジョニングプロファイルの配布用証明書には、チームの公開鍵しか入っていません。
- 配布用プロビジョニングプロファイル。配布用プロビジョニングプロファイルには、チームの配布用証明書とアプリケーションIDが入っています。プロビジョニングプロファイルがユーザテスト用（*アドホック配布用プロファイル*とも言う）であれば、アプリケーションIDで特定されるアプリケーションを、テスターが実行できるデバイスを識別するためにも使います。

配布用アセットを取得するには、次の手順を実行します。

1. 配布用証明書を取得する。
2. 配布しようとするアプリケーションの配布用プロファイルがチームになければ、これを生成する。
3. 配布したいアプリケーションの配布用プロファイルをダウンロードする。

次のステップ

- “App Storeに登録するためのプロジェクト設定” （22 ページ）
- “アプリケーションの配布” （42 ページ）

署名/プロビジョニングアセットの管理

この節では、もうすぐ期限切れになる、あるいはすでに期限切れになったプロビジョニングプロファイルを更新する手順、デジタルIDの安全を確保し、Mac間で共有する方法を説明します。

期限切れになった証明書の更新

開発用/配布用証明書が期限切れになった場合、Xcodeで新しい証明書を要求する必要があります。

© 2012 Apple Inc. All Rights Reserved.

もうすぐ期限切れになる、またはすでに期限切れになったプロビジョニングプロファイルの更新

プロビジョニングプロファイルの期限切れが近い、あるいはすでに期限切れになっている場合、更新が必要です。

その手順を以下に示します。

1. デバイスオーガナイザの「Library」セクションで、「Provisioning Profiles」を選択してください。
2. プロビジョニングプロファイルリストから、更新しようとするプロビジョニングプロファイルを選択します。
3. 「Renew」をクリックしてください。

更新したプロビジョニングプロファイルを、次の手順でデバイスにコピーします。

1. デバイスオーガナイザの「Library」セクションで、「Provisioning Profiles」を選択します。
2. 更新済みのプロビジョニングプロファイルを、リストからデバイス上にドラッグしてください。
3. デバイス側で、もうすぐ期限切れになる、またはすでになったプロビジョニングプロファイルを選択し、「Delete」をクリックして削除します。

署名/プロビジョニングアセットの保全と転送

署名/プロビジョニングアセットとは、開発デバイス上でiOSアプリケーションを実行するために用いる、秘密鍵、証明書、プロビジョニングプロファイルのことです。iOSアプリケーションの開発に用いるMacに、開発者のアセットが入っていない場合、当該アセットを移しかえる（転送する）必要があります。オーガナイザ上で、一方のMacから署名/プロビジョニングアセットをエクスポートし、もう一方にインポートしてください。

© 2012 Apple Inc. All Rights Reserved.

アプリケーションの設定

この章では、アプリケーションのエンタイトルメントを設定する方法について説明します。また、アプリケーションが動作するiOSリリース、デバイスファミリー、アーキテクチャを指定する方法も説明します。

このドキュメントの対象　以下の内容は、Xcode 4.3およびiOS SDK 5.0を対象としています。

iCloudエンタイトルメントの設定

iCloudエンタイトルメントには、ユーザがさまざまなデバイス上でアプリケーションを利用する場合に、iCloudストレージを使ってデータやドキュメントを共有できるようにする働きがあります。詳しくは"iCloud Storage"を参照してください。

アプリケーションがiCloudストレージを使うようにするためには、そのエンタイトルメントをオンにする必要があります。

XcodeはiCloudエンタイトルメントの値として、当該アプリケーションのバンドルIDを設定します。この値を変更しなければならない場合もあるので、「Configuring Your App's iCloud Entitlements」 in *iOS App Programming Guide* に従って確認してください。

アプリケーションが動作するiOSバージョンの指定

iOS（および、それに対応するSDK）の各バージョンには、以前のバージョンにはなかった機能が含まれています。iOSの新しいバージョンが公開されると、すぐにアップグレードするユーザもいれば、最新バージョンへの移行を延期するユーザもいます。アプリケーション開発の際、動作対象（ターゲット）とするiOSバージョンについては、次の2つの方針が考えられます。

- **最新のiOSバージョンをターゲットにする。**最新バージョンをターゲットにすると、最新バージョンのiOSで使用可能なすべての機能を利用できます。ただし、このアプローチでは、アプリケーションをデバイスにインストールできるユーザが少数になる可能性があります。なぜなら、ターゲットバージョンより前のiOSバージョンでは、このアプリケーションを実行できないからです。

- **以前のiOSバージョンをターゲットにする。**以前のバージョンをターゲットにすると、幅広いユーザ層にアプリケーションを提供できます（アプリケーションをターゲットのOSとそれ以降のバージョンで実行できるため）。ただし、アプリケーションが使用できるiOS機能が制限される可能性があります。

アプリケーションが動作するもっとも古いiOSバージョンを、次の手順で指定できます。

1. プロジェクトナビゲータで、該当するプロジェクトを選択してください。
2. プロジェクトエディタに列挙されている中から、アプリケーションの動作対象とするiOSリリースを選択してください。
3. 「Summary」をクリックします。
4. 「Deployment Target」ポップアップメニューから、動作対象とするiOSバージョンを選びます。

アプリケーションをビルドすると、配置ターゲットの選択がアプリケーションの`Info.plist`ファイル内の`MinimumOSVersion`エントリに反映されます。アプリケーションをApp Storeに投稿すると、このプロパティの値に基づいて、アプリケーションを実行できるiOSバージョンが表示されます。

> **注意** アプリケーションのビルドに使用したSDKが、アプリケーションのターゲットiOSバージョンよりも新しい場合（たとえば、ベースSDKはiOS 4.3、配布ターゲットはiOS 4.0の場合）、ターゲットiOSバージョンで利用できない機能がアプリケーションで使用されていることを検出すると、Xcodeはビルド警告を表示します。
>
> また、使用しているシンボルがアプリケーションの実行環境で利用できることを保証しなければなりません。利用できるか否かを確認する方法については、『*SDKCompatibilityGuide*』を参照してください。

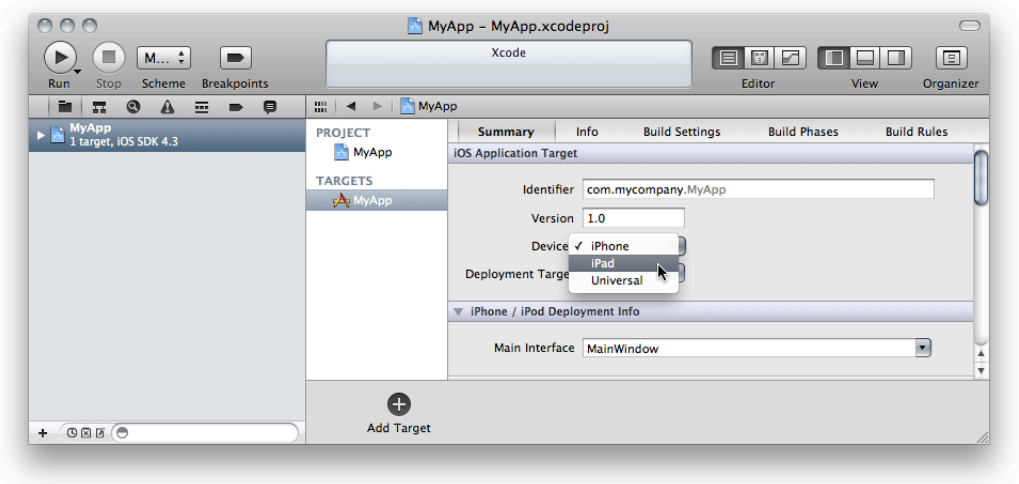
Important シミュレータ用にビルドしたバイナリは、ビルドに用いたXcodeツールセットに付属するバージョンのiOSシミュレータでしか動作しません。それより古いバージョンでも新しいバージョンでも動作しないのです。

アプリケーションが動作するデバイスの指定

「Devices」設定では、アプリケーションが動作するデバイスの種類を指定します。デバイスタイプにはiPhoneとiPadの2種類があります。iPhoneタイプには、iPhoneとiPod touchの両方のデバイスが含まれます。iPadタイプに含まれるのはiPadデバイスです。

アプリケーションを実行できるデバイスファミリーを指定するには以下の手順を実行します。

1. プロジェクトナビゲータで、該当するプロジェクトを選択してください。
2. プロジェクトエディタに列挙されている中から該当するものを選び、「Summary」をクリックします。
3. 「Devices」ポップアップメニューから、「iPhone」、「iPad」、「Universal」（両方で動作する）のいずれかを選択してください。



動作対象デバイス（iPhone/iPad/両方）の設定について詳しくは、“Advanced App Tricks”を参照してください。

アプリケーションが動作するアーキテクチャの指定

iOSのデバイスは、`armv6`および`armv7`を含むアーキテクチャセットのうちの1つを使用します。「アーキテクチャ(Architectures)」ビルド設定は、アプリケーションをビルドするためのアーキテクチャを指定します。この設定の値を指定する際には、次の2つの選択肢があります。

- **Standard。**サポートされているすべてのiOSデバイスと互換性のある、共通のアーキテクチャによってアプリケーションバイナリを生成します。このオプションは最小のアプリケーションを生成しますが、すべてのデバイスでできる限り高速に実行するための最適化は行われません。
- **Optimized。**サポートされているiOSデバイスごとに最適化されたアプリケーションバイナリを生成します。ただし、ビルド時間は「Standard」オプションを使用した場合よりも長くなります。また、複数の命令セットがバンドルされるためアプリケーションも大きくなります。

これらの事前定義の値によって提供されるものとは異なるアーキテクチャセットの実行可能コードを含むようにアプリケーションをビルドする場合は、「アーキテクチャ(Architecture)」ビルド設定の値リストから「その他(Other)」を選び、独自のiOSデバイスアーキテクチャの名前を入力します。

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

条件付きコンパイル/リンク

iOSアプリケーションは、シミュレータ上、実デバイス上の両方で動かすことができます。シミュレータを使ってデバイスの環境をシミュレートすれば、Mac上でアプリケーションの動作をテストできます。一方、実デバイス上で動かせば、実際の処理性能を確認できます。シミュレーション環境と実デバイス環境は基本的に異なります。そのため、2つの環境に別々に実装されたテクノロジーを使用する場合、一部は、シミュレータで実行しデバイス上で実行しないように、コードを調整する必要があります。

この節では、ビルドの目的に応じ、シミュレータ用/デバイス用のうちどちらのコードを生成するか、どのフレームワーク（やライブラリ）をリンクするか、を指定する方法を示します。

iOSアプリケーションのソースコードの条件付きコンパイル

デバイス上ではなくシミュレータ上でコードを実行する必要がある場合と、逆の場合があります。このような場合には、プリプロセッサマクロのTARGET_OS_IPHONEおよびTARGET_IPHONE_SIMULATORを使用して、あらゆるiOSベースのデバイスを対象に、コードを条件付きでコンパイルします。

リスト 2-1に、TARGET_IPHONE_SIMULATORマクロを使用して、iOSのためのコードをシミュレータ向けまたはデバイス向けにコンパイルすることを指定する方法を示します。

リスト 2-1　シミュレータ向けコンパイルの指定

```
// 変数helloに「Hello, <デバイスまたはシミュレータ>」という文字列を代入
#if TARGET_IPHONE_SIMULATOR
   NSString *hello = @"Hello, iOS Simulator!";
#else
   NSString *hello = @"Hello, iOS device!";
#endif
```

リスト 2-2は、Mac OS XとiOSで共用するソースにおいてTARGET_OS_IPHONEマクロを使用する方法を示します。

リスト 2-2　デバイス向けコンパイルの指定

```
#if TARGET_OS_IPHONE
   #import <UIKit/UIKit.h>
#else
   #import <Cocoa/Cocoa.h>
```

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

```
#endif
```

TARGET_OS_IPHONEマクロおよびTARGET_IPHONE_SIMULATORマクロは、TargetConditionals.hヘッダファイルに定義されています。

iOSアプリケーションのフレームワークの条件付きリンク

シミュレータ上で実行する場合と、デバイス上で実行する場合とで、異なるフレームワークにリンクするようにアプリケーションターゲットを設定する必要が生じることがあります。

特定のSDKを使用しているときにだけフレームワークをリンクするには、条件付きで、-framework <framework_name>に、定義を適用させるSDKのすべての構成で「ほかのリンカフラグ(Other Linker Flags)」ビルド設定を設定します。

必要であれば、「ほかのリンカフラグ(Other Linker Flags)」のビルド設定に別の条件を追加し、異なるSDKおよびフレームワークを指定できます。

ユーザテスト用アプリケーションへのiTunesアートワークの追加

テスト用アプリケーションには、iTunesがアプリケーションの識別に使用するアートワークを含める必要があります。そうしないと、ユーザがアプリケーションをiTunesライブラリに追加したときに、iTunesは汎用のアートワークを使用します。

テスターに表示されたiTunesアートワークが、アプリケーションのアイコンになります。このアートワークは、iTunesArtworkという名前の、512 x 512ピクセルのJPEGまたはPNGのファイルでなければなりません。このファイル名には拡張子を付けないでください。

ファイルをプロジェクトに追加するには、次の手順を実行します。

1. Xcodeでプロジェクトを開きます。
2. プロジェクトナビゲータで、該当するプロジェクトを選んでください。
3. 「File」＞「Add Files to “<App_Name>”」を実行します。
4. ファイルシステム上の、追加するファイルを選択します。
5. 「Copy items into destination group's folder (if needed)」を選択し、「Add」をクリックしてください。

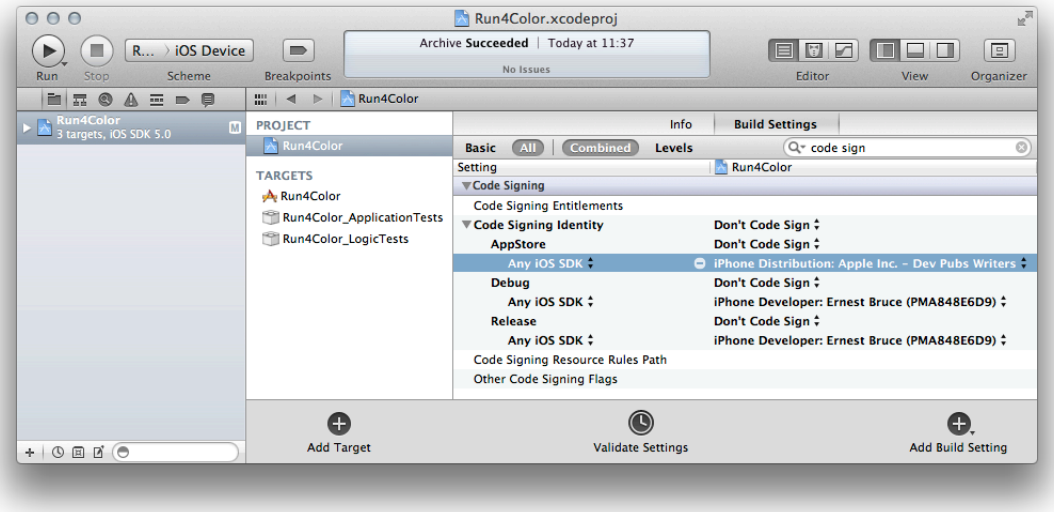
2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

App Storeに登録するためのプロジェクト設定

アプリケーションをApp Storeに登録し、公開するためには、ビルドの際、チームの配布コード署名IDが必要です。

アプリケーションに配布IDで署名し、それをアーカイブするよう、プロジェクトを次のように設定してください。

1. プロジェクトを開き、ビルド設定「Release」を「AppStore」という名前で複製します。
2. ビルド設定「Code Signing Identity」として、チームの配布コード署名IDを設定してください。これにより「AppStore」ビルド設定は、アプリケーションのDistributionプロビジョニングプロファイルを使うようになります。



3. ターゲット上で、「Release」ビルド設定の「Validate Build Product」ビルド設定仕様を削除してください。
4. アプリケーションをビルドするためのスキームを、「<App_Name>-AppStore」という名前で管理します。
5. 「AppStore」スキームで、「Archive」アクションのビルド設定を「AppStore」とします。

アプリケーションのビルドと実行

アプリケーションを実行してテストやデバッグをするためには、Xcodeのビルドシステムを使用してビルドする必要があります。ビルドエラーがなければ、シミュレータまたはデバイス上でアプリケーションを実行できます。

> **注意** 実際にユーザが使うことになる、iOSベースのデバイスモデル上でアプリケーションが正しく動作するよう、シミュレータ上だけでなく、当該デバイス上でもテストしなければなりません。
>
> デバイス上でアプリケーションを実行するには、開発チームのメンバになる必要があります。詳細については、"開発チームへの加入"（9 ページ）を参照してください。さらに、開発用のデバイスをXcodeに認識させることも必要です（"期限切れになった証明書の更新"（15 ページ）を参照）。

アプリケーションをビルドし実行する、一般的な手順は以下の通りです。

1. ビルド環境を指定する。
2. ビルド対象の動作環境（シミュレータ用かデバイス用か）を指定する。
3. アプリケーションをビルドする。
4. アプリケーションを実行する。

この章では、アプリケーションを実行するために必要な手順をそれぞれ説明します。iOSの機能を紹介したアプリケーションを参照する場合は、"サンプルアプリケーションの実行"（23 ページ）から始めてください。

> **このドキュメントの対象** 以下の内容は、Xcode 4.3およびiOS SDK 5.0を対象としています。

サンプルアプリケーションの実行

iOS Developer Libraryでは、iOSのアプリケーション開発プロセスについての学習に役立つさまざまなリソースを提供しています。これらのリソースタイプの1つにサンプルコードがあります。サンプルコードプロジェクトには、Xcodeのドキュメントオーガナイザ上で、あるいはウェブブラウザでiOS Dev

Centerを開いてアクセスできます。サンプルコードはシミュレータ上で実行可能です。開発チームのメンバーであれば、お使いのデバイス上でサンプルコードを実行することもできます。詳細については“開発チームへの加入”（9 ページ）を参照してください。

Xcodeを使ってサンプルコードを入手する手順は次の通りです。

1. 「Window」＞「Organizer」コマンドでXcodeオーガナイザを起動し、「Documentation」をクリックしてドキュメントオーガナイザを表示します。
2. ナビゲータ選択バーにある、「Browse」ボタン（「目」のアイコン）をクリックしてください。
3. 入手しようとするサンプルコードが収録されたライブラリを選択します。
4. 内容領域のテキストフィールドに「sample code」と入力します。
5. 内容領域にある、開きたいプロジェクトの名前をクリックしてください。
6. サンプルコードのプロジェクトページで、「Open Project」をクリックします。
7. プロジェクトの場所を選びます。

ウェブブラウザを使ってサンプルコードを入手する手順は次の通りです。

1. ウェブブラウザを起動し、http://developer.apple.com/devcenter/iosを開きます。「iOS Developer Library」というリンクをクリックしてください。
2. 左側にあるリストの「Resource Types」グループ以下にある、「Sample Code」をクリックします。
3. 「Documents」リストに並んでいる中から、入手したいプロジェクトの名前をクリックしてください。
4. 「Download Sample Code」をクリックします。

   プロジェクトディレクトリを収録したアーカイブがMac上にダウンロードされます。アーカイブは自動的に展開されます。されない場合はダブルクリックして展開してください。
5. サンプルコードのプロジェクトディレクトリを開いてください。
6. プロジェクトパッケージ（`.xcodeproj`という拡張子のファイル）をダブルクリックします。

   たとえば「HelloWorld」プロジェクトであれば、「`HelloWorld.xcodeproj`」というファイルをダブルクリックすることになります。Xcodeでプロジェクトが開きます。

   また、当該プロジェクトパッケージをDockのXcodeアイコンへドラッグしても、プロジェクトを開くことができます。

トラブルシューティング

- Xcodeが起動しない。

  Xcodeをダウンロードし、コンピュータにインストールします。その手順については、iOS Dev Centerにアクセスしてください。

Xcodeでサンプルコードプロジェクトを開いたら、次のセクションの手順に従ってアプリケーションをビルドして実行します。

「ビルドと実行」ワークフロー

このセクションでは、「ビルドと実行」のワークフローにおける各手順について説明します。

ビルド環境の設定

アプリケーションをビルドする際、Xcodeは、フレームワーク、ライブラリ、アプリケーション、コマンドラインツールなどのリソースで構成されるビルド環境を使用します。iOS SDKのリビジョンを公開するたびにこの環境の改良が行われ、ユーザインターフェイス機能の追加や、コンパイラやリソース処理ツールの強化がなされています。これらのリソースのほかに、アプリケーションのビルドをデバッグ用にするか配布用にするかを指定できます。このセクションでは、ビルド環境を設定する方法について説明します。

アプリケーションのビルドに用いるSDKの設定

Xcodeでのアプリケーションのビルド方法を決定する主な要素の1つは、ビルドに使用するSDKです。

アプリケーションのビルドに用いるSDKの指定は、ビルド設定「Base SDK」を設定して行います。

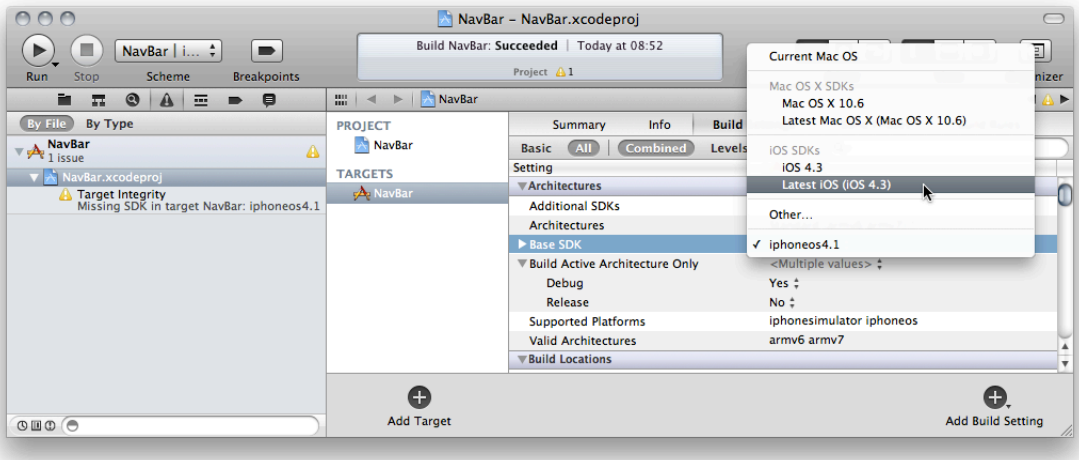
> **注意** 新しいSDKリリースを採用したときのプロジェクトの再構成を最小限に抑えられるように、プロジェクトの「ベースSDK(Base SDK)」を特定のSDKリリースではなく、「最新のiOS(Latest iOS)」に設定します。こうすると、プロジェクトは常に、Xcodeツールセットに含まれている最新のSDKを使用します。

Base SDK Missing（ベースSDKがない）

プロジェクトの「ベースSDK(Base SDK)」の設定が特定のiOS SDKリリースに設定されていると、そのSDKリリースを利用できない以降のXcodeツールセットで同じプロジェクトを開いたときに、「ベースSDK(Base SDK)」に有効な値が設定されません。この場合、イシューナビゲータに、解決を要する問

題として「Missing SDK」という項目が現れます（図 3-1を参照）。この問題は、ターゲットの「ベースSDK(Base SDK)」を、利用可能なSDKリリース、または「最新のiOS(Latest iOS)」に設定することにより解決できます。

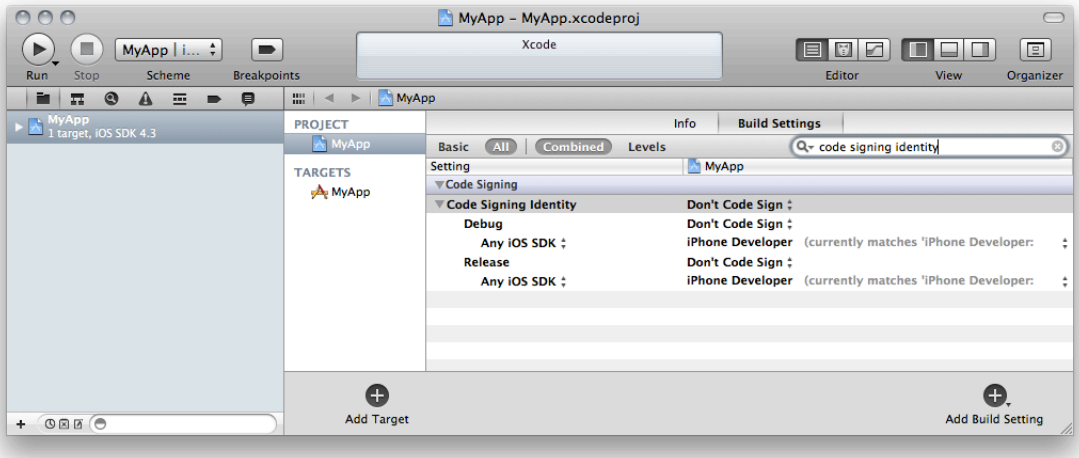
図 3-1　SDKが欠落しているプロジェクトの様子



コード署名IDの設定

デバイス上で実行するようにアプリケーションをビルドする場合は、キーチェーンに格納されている開発用証明書（コード署名IDとも呼ばれます）を利用してアプリケーションに署名します。開発用証明書を取得してインストールする方法については、“開発用デバイスの設定”（10 ページ）を参照してください。

「コード署名ID(Code Signing Identity)」ビルド設定は、Xcodeがバイナリへの署名に使用する、プロビジョニングプロファイルとコード署名IDを指定します。Xcodeは、デフォルトのキーチェーンからコード署名IDを検索します。

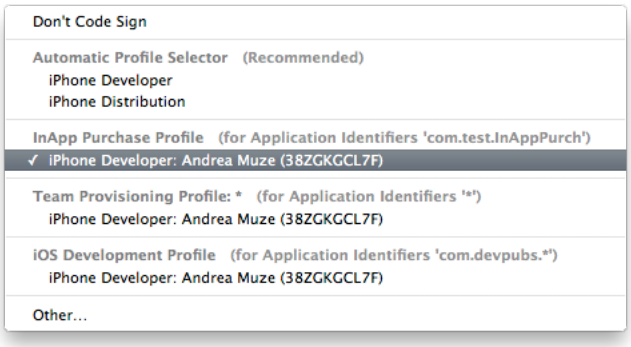


「コード署名ID(Code Signing Identity)」ビルド設定に設定可能な値は、次の通りです。

- **Don't Code Sign**。シミュレータ用にのみビルドします。
- **Automatic Profile Selector**。「iPhone Developer」または「iPhone Distribution」で始まる名前を持つプロビジョニングプロファイルが選択されます。
- **Specific Profile**。アプリケーションが特別なエンタイトルメントを要する場合に、該当するプロビジョニングプロファイルのコード署名IDを選択します（"特殊な開発用にデバイスを設定する"（13 ページ）を参照）。期限切れまたはその他の理由で無効になったプロビジョニングプロファイルは、グレイ表示されており使用できません。

  図3-2に、ビルド設定「Code Signing Identity」として、特別な開発用のプロビジョニングプロファイルを選択した様子を示します。

図 3-2　コード署名IDとして特別なプロビジョニングプロファイルを選択している様子



プロジェクトテンプレートは、自動セレクタ（Automatic Profile Selector）で署名IDを設定するようになっています。アプリケーションが特別なプロビジョニングプロファイルを使う場合に限り、ビルド設定「Code Signing Identity」の値を変更する必要があります。詳しくは"期限切れになった証明書の更新"（15 ページ）を参照してください。

Important 同じ名前の別のコード署名IDを使用する必要がある場合は、IDごとに別のMac OS Xユーザアカウントを使用しなければなりません。

実行環境の指定

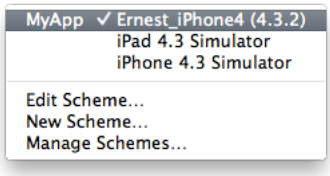
アプリケーション実行環境に関しては、（1）アプリケーションが実行される位置、（2）サンドボックスに置くアプリケーションデータ、（3）Core Locationフレームワークが報告するロケーション/トラック、の3つを意識する必要があります。

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

実行デスティネーションの指定

アプリケーションをビルドする前に、どの環境で動くコードをビルドするのか、Xcodeに設定する必要があります。この*実行デスティネーション*の指定は、ツールバーの「Scheme」メニューで行います。このメニューを使って、たとえばデバイスによるアプリケーションのパフォーマンスをテストするために、実行環境をシミュレータからデバイスに切り替えてアプリケーションを実行することができます。

有効なプロビジョニングプロファイルを組み込んだデバイスをMacに接続すると、その名前と動作しているiOSバージョンが、ツールバーの「Scheme」メニューに選択肢として現れます。このメニューで、アプリケーションをデバイス上で動かすか、シミュレータで動かすかを切り替えます。



トラブルシューティング

- “実行デスティネーションとしてデバイスが表示されない”（54 ページ）。
- “デバイスを接続したとき、Xcodeに「Unknown iOS Detected」というダイアログが表示される”（55 ページ）。

アプリケーションデータの指定

スキームの「Run」アクションは、アプリケーション実行前にXcodeがサンドボックスに置くデータに影響を与えます。単体テスト、すなわち、アプリケーションコード中の重要なユニットについて動作が正しいかどうか確認するテストでは、その目的のために用意したアプリケーションデータアーカイブを使うと有用です。あらかじめ中身が分かっているデータセットを使ってテストすれば、詳細な単体テストが可能であり、テスト自身の中でテストデータを設定する必要もありません。

注意 指定できるアプリケーションデータアーカイブは、スキームごとに1つだけです。複数のアプリケーションデータアーカイブを常に使い分ける必要がある場合は、アーカイブごとにスキームを定義してください。

Important 「Run」アクションでは、プロジェクトの一部として保存したアプリケーションデータアーカイブしか指定できません。詳しくは"アプリケーションデータを開発デバイスからコピーするには"（31 ページ）を参照してください。

アプリケーションの実行前に、サンドボックスに置くデータを指定するには：

1. 「Scheme」ツールバーメニューから、使用するスキームを選択してください。
2. 「Run」アクションを選択してください。
3. 「Application Data」ポップアップメニューから、使用したいアプリケーションデータアーカイブを指定します。

ロケーションまたはトラックの指定

ロケーションベースのアプリケーションでは、起動時や実行中に実行デスティネーションがアプリケーションに報告する、ロケーションやトラックを指定できます。

GPS eXhange Format（GPX）ファイルに、ある特定のロケーション（単一の中間地点）またはトラック（中間地点の順序つきコレクション）を指定します。トラックをシミュレートする場合、シミュレータやデバイスは、トラックに指定された順序で中間地点を報告します。

プロジェクトでGPXファイルが使えるようにするためには、既存のファイルがあればこれをプロジェクトに追加し、ない場合は新規に作成した上で、ロケーションまたはトラックの詳細を指定してください。

アプリケーションのビルド

ビルドプロセスを起動するには、「Product」＞「Build」を選びます。「Build」がグレイ表示になっている場合は、正しい実行デスティネーションを選択してください（"実行デスティネーションの指定"（28 ページ）を参照）。

アクティビティビューア（ワークスペースウインドウのツールバーの中央）に、ビルドの進捗状況と問題の有無が表示されます。ビルドに失敗したり警告が現れたりした場合は、次の手順でログビューアを開き、状況を調べてください。

1. 「View」＞「Navigators」＞「Log」を実行して、ログナビゲータを開きます。
2. 左側のリストから、調べたいビルドタスク名を選択してください。

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

編集領域のログビューアに、ビルドの過程で実行した処理が列挙されます。

ビルドが正常に完了すれば、アプリケーションを実行できるようになります（“アプリケーションの実行”（30 ページ）を参照）。

トラブルシューティング
- “Xcodeがアプリケーションを開発用デバイスにインストールできない”（51 ページ）
- “プロビジョニングプロファイルが期限切れになった”（52 ページ）
- “コード署名IDのビルド設定が、キーチェーンの有効なコード署名IDを識別できない。”（53 ページ）
- “Xcodeが証明書を信頼しない”（53 ページ）
- “キーチェーンにコード署名IDが重複して登録されている”（53 ページ）
- “プロビジョニングプロファイルのアプリケーションIDが、アプリケーションのバンドルIDと一致しない”（54 ページ）

アプリケーションの実行

アプリケーションを実行するには、「Product」＞「Run」を選びます。

アプリケーションを実行すると、Xcodeはシミュレータ環境またはデバイス上にアプリケーションを配備し、起動します。

デバイス上でアプリケーションが動作すれば、デバイスのすべての機能を使用して、想定通りに動作していることを確認できます。特に、アプリケーションがデバイスのリソース（CPU、メモリ、バッテリーなど）を可能な限り効率的に使用することを確認すべきです。詳細については、“アプリケーション性能の最適化”（39 ページ）を参照してください。

注意 デバイス上でアプリケーションを実行するためには、デバイスがUSBケーブルでMacに接続されていなければなりません。

アプリケーションデータの管理

デバイス上やシミュレーション環境に初めてアプリケーションをインストールする時点で、iOSはサンドボックス（アプリケーションのホームディレクトリとも呼ぶ）を生成します。「File and Data Management」 in *iOS App Programming Guide* で説明されている通り、iOSアプリケーションはサンドボックス内に存在するファイルにのみアクセスできます。

Xcodeは、アプリケーションを再インストールする際、アプリケーションデータを削除しません。ただし、ユーザが使用するときと同じ状態でアプリケーションをテストする場合は、そのような情報を削除する必要があるかも知れません。それには、ホーム（Home）画面からアプリケーションを削除します。詳細については、“アプリケーションのアンインストール”（35 ページ）を参照してください。

単体テストの際など、サンドボックス内のアプリケーションデータを既知のものに置き換え、所定の条件のもとでテストしたい場合もあるでしょう。そのためにはまず、開発デバイス上で、アプリケーションデータのアーカイブを作成します（シミュレータ上では生成できません）。いったんアーカイブができてしまえば、Mac上でそのアプリケーションデータを修正し、デバイスに書き戻すことも可能です。

アプリケーションデータを開発デバイスからコピーするには

1. 当該データを使うアプリケーションが組み込まれたデバイスを接続してください。
2. デバイスからアプリケーションデータをコピーします。

アプリケーションデータをファイルシステム上にコピーすると、たとえばアプリケーション自身の上では手間がかかるような変更でも、容易に施すことができます。アプリケーションデータアーカイブの中身にアクセスしたい場合は、Finder上でControlキーを押しながらアーカイブをクリックし、「Show Package Contents」を実行してください。

アプリケーション実行時に所定のアプリケーションデータを使いたい場合は、適当なアプリケーションデータアーカイブをプロジェクトに追加し、アプリケーションスキームの「Run」アクションでこのアーカイブを指定します。詳細については、“アプリケーションデータの指定”（28 ページ）を参照してください。

アプリケーションデータを開発デバイスにコピーするには

1. 当該データを使うアプリケーションが組み込まれたデバイスを接続してください。
2. デバイスにアプリケーションのデータをコピーします。

アプリケーションのシミュレーション環境サンドボックスにアクセスするには、Finderで`~/Library/Application Support/iPhone Simulator/<sdk_version>/Applications`ディレクトリに移動します。次に、アプリケーションのバイナリファイルを見つけるには、`Applications`ディレクトリ内の各ディレクトリを開きます。バイナリファイル以外にも、`Documents`、`Library`など、アプリケーションのサンドボックスを構成するディレクトリが存在します。

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

さらに学びたい方へ

Xcodeを使用してアプリケーションをビルドおよび実行する方法の詳細については、“Building and Running Your Code”を参照してください。

iOSシミュレータの使用

iOSシミュレータを使えば、開発したiOSアプリケーションがMac上で動作します。アプリケーションの動作をシミュレートすることにより、次のようなことができます。

- 開発チームのメンバーになる前に、Xcodeでの開発方法やiOSの開発環境について学ぶことができる。
- 設計や初期のテストの間に、アプリケーションの主な問題点を発見できる。
- アプリケーションのユーザインターフェイスをテストできる。
- iOSベースのデバイス上で詳細なパフォーマンス分析を実行する前に、アプリケーションのメモリ使用量を測定できる。

iOSシミュレータ（実体は`<Xcode>/Platforms/iPhoneSimulator.platform/Developer/Applications`以下）は、iPhoneまたはiPadのユーザインターフェイスをコンピュータ上のウインドウに表示します。このアプリケーションは、キーボードとマウスを使用してタップやデバイスの回転、その他のアクションをシミュレートすることにより、さまざまな方法でコンピュータとやり取りできます。

この章では、コンピュータの入力デバイスを使用して、ユーザとデバイス間のやり取りをシミュレートする方法を説明します。また、シミュレーション環境からアプリケーションをアンインストールする方法およびシミュレータのコンテンツをリセットする方法も示します。

> **このドキュメントの対象**　以下の内容は、Xcode 4.3およびiOS SDK 5.0を対象としています。

デバイスとiOSバージョンの設定

iOSシミュレータは、3種類のデバイス（iPhone、Retinaディスプレイ付きのiPhone、iPad）と、数種のバージョンのiOSをシミュレートできます。

シミュレートするデバイスを指定するには、「ハードウェア(Hardware)」＞「デバイス(Device)」を選んでデバイスを選びます。

シミュレートするiOSバージョンを指定するには、「ハードウェア(Hardware)」＞「バージョン(Version)」を選んでiOSバージョンを選びます。

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

ハードウェアの操作

iOSシミュレータを利用すると、ユーザがデバイス上で実行するほとんどの動作をシミュレートできます。アプリケーションがシミュレータ上で動作している間、「ハードウェア(Hardware)」メニューで以下のハードウェア操作を実行できます。

- **反時計回りに回転(Rotate Left)。**シミュレータを左に回転します。
- **時計回りに回転(Rotate Right)。**シミュレータを右に回転します。
- **シェイクジェスチャ(Shake Gesture)**シミュレータをシェイクします。
- **ホーム(Home)。**シミュレータをホーム(Home)画面に戻します。
- **ロック(Lock)。**シミュレータをロックします。
- **メモリ警告をシミュレート(Simulate Memory Warning)。**最上位のアプリケーションにメモリ不足の警告を送信します。メモリ不足状態の対処方法については、「Observing Low-Memory Warnings」in *iOS App Programming Guide*を参照してください。
- **着信ステータスバーを切り替える(Toggle In-Call Status Bar)。**ステータスバーを、通常状態と、電話やFaceTimeの着信状態との間で切り替えます。ステータスバーは、通用状態よりも着信状態の方が背が高くなります。このコマンドは、着信中にユーザがアプリケーションを起動すると、アプリケーションのユーザインターフェイスがどうなるかを示します。
- **ハードウェアキーボードをシミュレート(Simulate Hardware Keyboard)。**iPadシミュレータのソフトウェアキーボードを切り替えます。iPadデバイスでKeyboard DockまたはWireless Keyboardを使用してシミュレートするにはソフトウェアキーボードをオフにします。
- **TV出力（TV Out）。**デバイスのTV出力信号をシミュレートするウインドウを開きます。

ジェスチャの実行

表4-1に、シミュレータ上で実行できるジェスチャを示します（ジェスチャについての情報は、『*iOS Human Interface Guidelines*』を参照してください）。

表 4-1　iOSシミュレータでのジェスチャの実行

ジェスチャ	デスクトップアクション
タップ	クリック。
タッチアンドホールド	マウスボタンを押したままにします。
ダブルタップ	ダブルクリック。

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

ジェスチャ	デスクトップアクション
スワイプ	1. スワイプを開始したい位置にポインタを置きます。 2. マウスボタンを押したままにします。 3. スワイプしようとする方向にポインタを移動し、マウスボタンを放します。
フリック（はじく）	1. ポインタを開始位置に置きます。 2. マウスボタンを押したままにします。 3. すばやくフリックしようとする方向にポインタを移動し、マウスボタンを放します。
ドラッグ	1. ポインタを開始位置に置きます。 2. マウスボタンを押したままにします。 3. ドラッグしようとする方向にポインタを移動します。
ピンチ	1. ピンチを発生させようとする位置にポインタを置きます。 2. Optionキーを押したままにします。 3. 指でのタッチを表す円を開始位置に移動します。 4. Shiftキーを押したまま、円を目的の中心位置まで移動してからShiftキーを放すことにより、ピンチターゲットの中心点を移動します。 5. マウスボタンを押したまま、その円を終了位置まで移動し、Optionキーを放します。

アプリケーションのインストール

シミュレータ用にアプリケーションをビルドすると、Xcodeは自動的にシミュレーション環境にアプリケーションをインストールします。詳細については、“アプリケーションのビルドと実行”（23 ページ）を参照してください。

> **注意** App Storeから入手したアプリケーションをシミュレーション環境にインストールすることはできません。

アプリケーションのアンインストール

シミュレーション環境にインストールしたアプリケーションをアンインストールするには、デバイスからアプリケーションをアンインストールするのと同じ方法を使用します。

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

1. アンインストールしたいアプリケーションのアイコンの上にポインタを置き、アイコンが揺れ始めて「閉じる」ボタンが現れるまでマウスボタンを押し続けます。
2. 「Close」ボタンを押してください。
3. 「ホーム(Home)」ボタンをクリックして、アイコンが揺れるのを停止します。

コンテンツと設定のリセット

シミュレーション環境のユーザコンテンツと設定を工場出荷時の状態に戻し、インストールしたアプリケーションを削除するには、「iOSシミュレータ(iOS Simulator)」＞「コンテンツと設定をリセット(Reset Content and Settings)」を選びます。

iOSシミュレータのコンソールログとクラッシュログの表示

アプリケーションをシミュレータで実行する際にコンソールログを表示する方法については、“コンソール出力ログとデバイスログの表示” （39 ページ）を参照してください。

シミュレータ上で動作しているアプリケーションがクラッシュすると、CrashReporterにその詳細が表示されます。クラッシュ時のCrashReporterの動作は、CrashReporterPrefアプリケーション（`<Xcode>/Applications/Utilities`以下）で設定できます。ここで`<Xcode>`は、Xcodeツールセットのインストール先ディレクトリです。

シミュレーション環境におけるファイルシステムの位置

iOSシミュレータがシミュレートできるiOSリリースのファイルシステムは、ホームディレクトリ（`~/Library/Application Support/iPhone Simulator`）に格納されます。このディレクトリには、iOSシミュレータによってサポートされているiOSリリースごとに、1つのディレクトリが含まれています。

iOSリリースごとの各ディレクトリ内で、システムアプリケーションの環境設定ファイルは`Library/Preferences`に、他社製アプリケーションの環境設定ファイルは`Applications/<app_UUID>Library/Preferences`に格納されています。

ハードウェアシミュレーションサポート

iOSシミュレータは加速度計やカメラハードウェアのシミュレートは行いません。

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

高い品質と性能の保証

この章では、アプリケーションのコードが想定通りに動作し、リソースを適切に使い、高速に動くようにするための、さまざまな方針や機構について説明します。

このドキュメントの対象　以下の内容は、Xcode 4.3およびiOS SDK 5.0を対象としています。

コードの正しさの保証

単体テストによりアプリケーションの機能をテストすると、堅牢でセキュアなコードを書くために役立ちます。Xcodeは使いやすく柔軟な単体テスト環境を提供しており、そのテスト環境を使うことによりコードに変更が重ねられてもそのコードを確実に設計通りに動作するようにできます。

単体テストには、ロジックテストとアプリケーションテストの2種類があります。**ロジックテスト**では、最下層レベル（通常はメソッドや単独のクラス）でコードが正しく動作することを確認します。一方**アプリケーションテスト**では、アプリケーションの各クラスが連携して、想定通りに動作することを確認します。

Xcodeでは、単体テストにも対応したプロジェクトやプロダクトを容易に作成できます。また、既存のプロジェクトやプロダクトに、後から単体テストを追加することも可能です。

プロジェクトに単体テストを統合する方法については、『*Xcode Unit Testing Guide*』を参照してください。

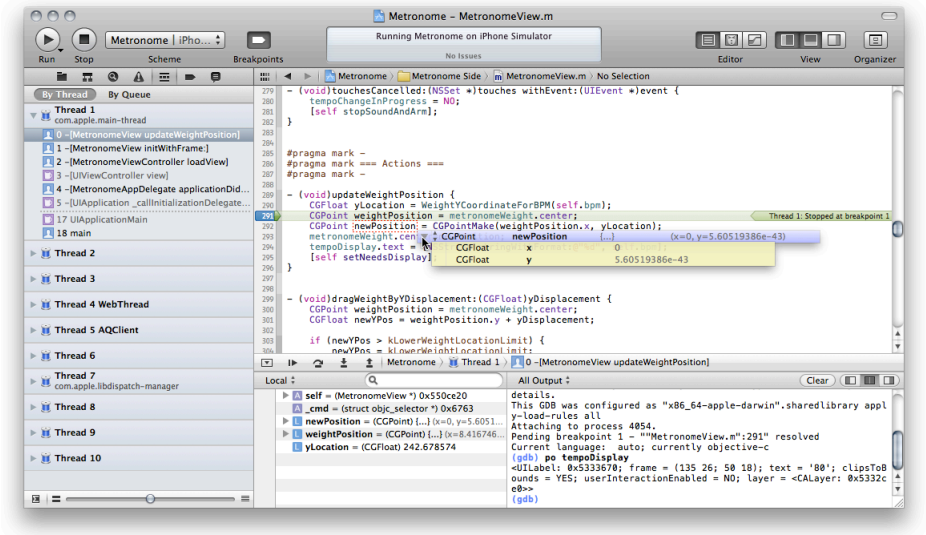
バグの修正

この節では、Xcodeのデバッグ機能概要と、コンソール出力の見方を説明します。

デバッグのワークフローにおいて、アプリケーションがサンドボックスに書き込んだデータを表示または操作する必要が生じることがあります。たとえば、テストのための特定の条件を再現できるよう、アプリケーションが保存したデータを編集する場合があります。アプリケーションのデータの操作方法についての詳細は、“アプリケーションデータの管理” （30 ページ）を参照してください。

デバッグ機能の概要

Xcodeにはさまざまなデバッグ機能が統合されており、必要に応じて、アプリケーション全体を俯瞰し、あるいは細部を調査することができます。



- **デバッグナビゲータ。**このナビゲータ（ワークスペースウインドウの左側）には、アプリケーションのスレッドやスタックフレームが表示されており、動作中のコードを調べることができます。いずれかの項目を選択すると、該当するファイルがソースエディタ上に開きます。
- **ソースエディタ。**ソースエディタ（編集領域内）を利用すると、まさにコード内でコードをデバッグできます。ここでは、ほとんどのデバッグ機能が利用できます。
  - ブレークポイントの設定
  - 変数の上にポインタを置くことによる、変数値の表示
  - 指定した行まで実行
  - 関数やメソッドの呼び出しへのステップイン、ステップアウト、ステップオーバ
    ▼altDescriptionForタグの訳：Xcodeのインラインデバッグ▼
- **デバッグ領域。**デバッグ領域（ソースエディタの下側）には、変数リストと、プログラムのコンソール出力を表示するコンソールペインがあります。コンソールペインにデバッグコマンドを入力し、実行することも可能です。デバッグバー（デバッグ領域の一番上）では、プログラムを停止する、続行する、ステップスルーするなどの操作ができます。

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

Important デバイス上でうまくアプリケーションをデバッグするためには、プロジェクトで使っているiOS SDKのバージョン番号が、デバイスのiOSバージョンのバージョン番号以降でなければなりません。Xcodeの最新の公開バージョンをApp Storeから、またはベータバージョンをhttp://developer.apple.com/devcenter/iosから入手し、インストールするとよいでしょう。

コンソール出力ログとデバイスログの表示

UIKitなどのiOSフレームワークは、予期せぬイベントが発生すると、それを示すためにコンソールにログエントリを送信します。また、iOSアプリケーション内でコンソールメッセージを送出することもできます。コンソールメッセージを送出する方法として、`NSLog`関数を使用する方法があります。コンソールログも、アプリケーションのロジックを解析したり、バグを追跡するのに役立ちます。

シミュレータでアプリケーションを実行している場合は、コンソールアプリケーション（`/Applications/Utilities`にあります）でコンソールログにアクセスできます。デバイスでアプリケーションを実行する場合、アプリケーションからのログエントリはXcodeの「オーガナイザ(Organizer)」に表示されます。

デバイスのコンソール出力を表示するには、次の手順を実行します。

1. 「ウインドウ(Window)」＞「オーガナイザ(Organizer)」を選び、オーガナイザ(Organizer)ウインドウを開きます。
2. 「デバイス(Devices)」をクリックして、デバイスオーガナイザを開きます。
3. ログを表示したいデバイスのセクションで、「Device Logs」を選択してください。

「オーガナイザ(Organizer)」の「Device Logs」ペインには、アプリケーションのクラッシュについての情報が含まれています。クラッシュリストをリフレッシュするには、デバイスを一度はずして再接続しなければならない場合があります。

クラッシュログの詳細については、「Understanding and Analyzing iPhone OS Application Crash Reports」を参照してください。

アプリケーション性能の最適化

アプリケーション性能の最適化は、開発工程における重要な部分です。最適化は特に、iOSベースのデバイスで重要です。これは強力なコンピューティングデバイスですが、デスクトップコンピュータやポータブルコンピュータが持つメモリやCPUパワーは持ち合わせていません。また、ユーザのバッテリー持続時間に直接影響を与えるため、アプリケーションのバッテリー使用にも注意を払う必要があります。

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

この節では、アプリケーションのパフォーマンスを測定したり調整するために使用する、Xcodeのグラフィカルツール Instrumentsを紹介します。

一般的なパフォーマンスのガイドラインについては、“Performance Tuning”を参照してください。

instrumentアプリケーション

Instrumentsアプリケーションを利用すると、メモリやネットワークの使用状況などのアプリケーションのパフォーマンス測定値を収集できます。シミュレータまたは開発デバイス上で実行中のiOSアプリケーションからデータを収集できます。

ユーザが満足できる体験を提供するには、iOSアプリケーションが、デバイスのリソースをできる限り効率的に使用することが重要です。たとえば、ユーザに動作が遅いと感じさせたり、バッテリーがすぐに消費されてしまうと感じさせるようなリソースの使い方をしてはいけません。メモリを使い過ぎるアプリケーションは、動作が遅くなります。動作するためにネットワークアクセスに依存するアプリケーションは、できる限りネットワークの使用頻度を下げる必要があります。ネットワーク通信用の無線を駆動すると大量のバッテリーを消費するからです。

Instrumentsは高度なデータ収集インターフェイスを備えており、CPU、メモリ、ファイルシステムなど、アプリケーションによるリソースの使用状況を正確に把握できます。

Instrumentsは、ソフトウェアベースのデータ収集ツール（*instrument*とも呼ばれる）を使用してパフォーマンスデータを収集します。**instrument**は、ネットワーク活動やメモリ使用状況など、特定の種類のデータを収集します。**トレースドキュメント**には、1つのアプリケーションに関するデータを収集するinstrumentが1つ以上が含まれています。

iOSアプリケーションの多くはシミュレータ上で動作し、設計上の判断事項をここでテストできますが、デバイスにおける動作と完全に同じというわけではありません。たとえばCPU速度やメモリスループットなど、デバイスの性能特性は再現されていないのです。ユーザが実際に用いるデバイス上でアプリケーションのパフォーマンスを効果的に測定するには、実際のデバイスを使用する必要があります。（プロセッサ速度、メモリ制限、特殊なハードウェアなどの点から見て）実際のデバイスの実行環境を正確に表すものを取得できるのは、デバイス上でのみになります。

シミュレータ上ではテストできず、デバイス上で実行しなければならない事項には、次のようなものがあります。

- **3本以上の指で起こすイベント。**
- **加速度計の実際の読み取り値。**UIKitフレームワークを介して、コンピュータの加速度センサーにアクセスできますが（加速度センサーがある場合）、その測定値と、デバイス上の加速度センサーの測定値は異なります。その差違は、コンピュータとiOSベースのデバイスでは、ほかのハードウェアに対する画面の相対的な位置が異なることが大きな原因です。

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

- **OpenGL ESによるレンダリング。**OpenGL ESはデバイス上のレンダラを使用します。これは、シミュレータで使用しているものとは若干異なります。このため、シミュレータ上の場面とデバイス上の同じ場面では、ピクセルレベルで同じでない場合があります。詳細については、「Drawing with OpenGL ES」 in *iOS App Programming Guide*を参照してください。

デバイス上でアプリケーションのパフォーマンスを測定するには、次の手順を実行します。

1. “アプリケーションのビルドと実行”（23 ページ）の説明に従って、デバイス上でアプリケーションをビルドして実行します。
2. アプリケーションを停止します。
3. Instrumentsを起動します。

   Instrumentsアプリケーションは`<Xcode>/Applications`にあります（`<Xcode>`はXcodeツールセットのインストール場所を示します）。
4. テンプレート（「Activity Monitor」など）を選んで、トレースドキュメントを作成します。
5. 「Target」ツールバーメニューから、パフォーマンスデータの収集対象となるアプリケーションを含むデバイスを選択します。
6. トレースドキュメントにinstrumentを追加または削除して、必要なデータを収集します。
7. 「Target」ツールバーメニューから、起動するアプリケーション（手順1で実行したものと同じアプリケーション）を選択します。
8. 「Record」をクリックするとデータ収集が始まります。
9. アプリケーションの、テストしたい機能を使ってみてください。

アプリケーションパフォーマンスの測定および分析の詳細については、『*Instruments User Guide*』を参照してください。

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

アプリケーションの配布

App Storeを通して、ユーザテスト用または一般配布用にアプリケーションを共有する準備が整ったら、Distributionプロビジョニングプロファイルを使用してアプリケーションのアーカイブを作成し、アプリケーションのテスターに送付するかiTunes Connectに投稿する必要があります。この章ではこれらの作業の進めかたを説明します。

アプリケーションを配布するためには、“開発用デバイスの設定”（10 ページ）に示した要件を満たさなければなりません。

このドキュメントの対象　以下の内容は、Xcode 4.3およびiOS SDK 5.0を対象としています。

ユーザテスト用アプリケーションの公開

自分自身やチームメイトの支援によって、アプリケーションのテストとチューニングが終了したら、アプリケーションを使用する可能性がある代表的なサンプルユーザに、より広範囲なテストを実施してもらうのが良いでしょう。このようなテストによって、特定の使用パターンでのみ発生する問題が明らかになる場合があります。**アプリケーションテスター**とは、開発チームには属していないけれども、App Storeでリリースされる前にアプリケーションをテストしたいと考える潜在ユーザです。アプリケーションテスターを参加させると、アプリケーションはさまざまな使い方をされることになります。クラッシュレポート（あるいはクラッシュログ）を収集、分析して、問題解決に役立てることができます。

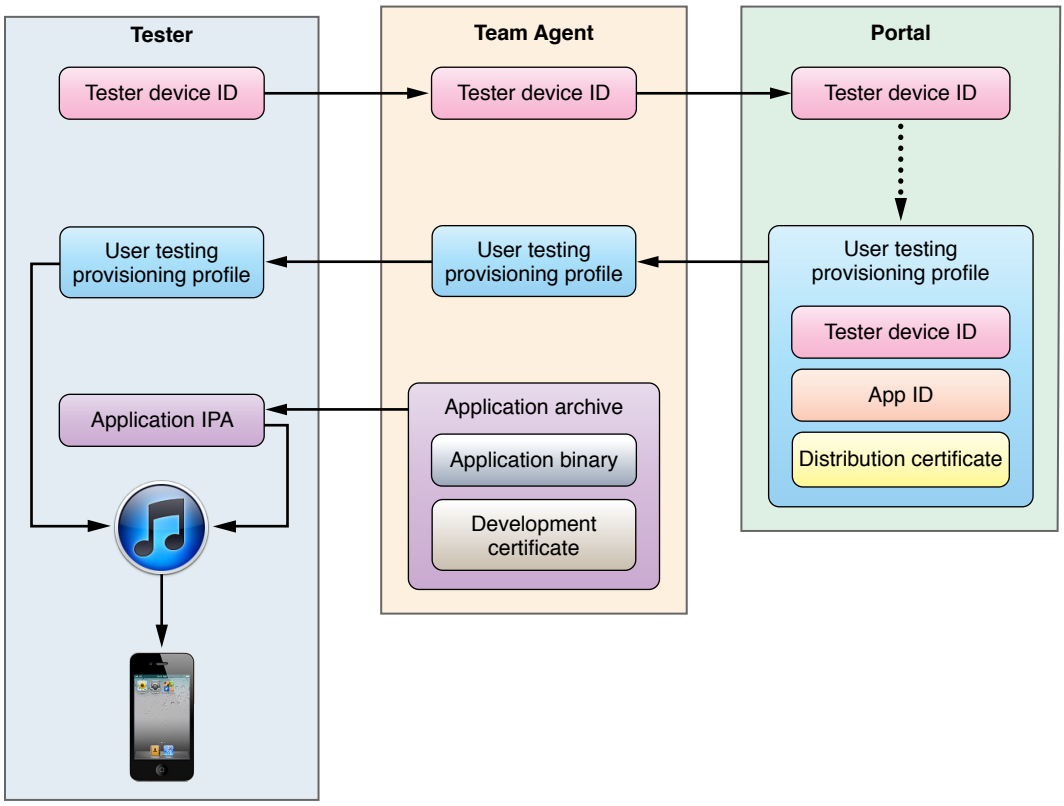
Important　ユーザテスト用にアプリケーションを公開できるのは、チーム内の権限ある1人だけです。

開発期間中、iOSアプリケーションを実際に動かしてみることができるのは、所定の開発者、デバイス、アプリケーションを設定した、プロビジョニングプロファイルがインストールされたデバイス上に限ります。チームメイト（開発者自身を含む）は、デバイスを開発用にプロビジョニングする際に、このファイルをデバイスにインストールします（“開発アセット、配布アセットの設定”（9 ページ）を参照）。開発チーム以外のユーザをテストに追加するには、ユーザのデバイスをチームに追加して、**ユーザテスティングプロビジョニングプロファイル**（暫定プロビジョニングプロファイルとも呼ばれる）を発行する必要があります。これによって、チームに属していないユーザもアプリケーションをデバイスにインストールできます。

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

図 6-1に、ユーザテスト用にアプリケーションを公開する際に必要な事項を示します。

図 6-1 テスターにアプリケーションを配布する際に必要な事項



テストグループにテスターを追加するために必要な情報をユーザに入手してもらい、テスターにクラッシュログの送信方法を知らせるには、“アプリケーションのテスター向けの説明書”（45 ページ）の情報をテスターに送信します。

> **ユーザテストの要件** ユーザテスト用にアプリケーションを公開するためには、iTunesアートワークが必要です。“ユーザテスト用アプリケーションへのiTunesアートワークの追加”（21 ページ）を参照してください。

このセクションの残りのセクションでは、自分のチームにテスターを追加する手順を説明し、テスターから送られたクラッシュログを「オーガナイザ(Organizer)」にインポートする方法を示します。

チームへのユーザテスト用デバイスの追加

ユーザのデバイスを、ユーザテスト用にチームに追加するには、以下の手順を実行します。

1. ユーザデバイスのIDを取得してください。

この情報を得るには、電子メールを利用するのがもっとも簡単な方法です。“デベロッパへのIDの送信”（45 ページ）に説明されている、デバイスIDを送信する手順をテスターに実行してもらいます。

2. ユーザのデバイスIDをチームに追加します。

> **ヒント** テスターの電子メールアドレスをデバイス名として使用します。

ユーザテスト用のプロビジョニングプロファイルの設定

iOSプロビジョニングポータルで、すでにアプリケーションのユーザテスト用プロビジョニングプロファイルがあれば、テスターのデバイスIDをそこに追加します。そうでなければ、次の特性でDistributionプロファイルを作成します。

配布メソッド	暫定
プロファイル名	<App_Name>ユーザテスト用プロファイル
アプリケーションID	テスト対象アプリケーションの適切なアプリケーションID
デバイス	テスターのデバイスID

プロビジョニングプロファイルの名前に、アプリケーション名を含めるか否かは任意です。

アプリケーションIDがあれば、個々のアプリケーション、あるいはアプリケーションドメインを識別できます。

テスターへのアプリケーションの送信

テスターにアプリケーションを送信するには、次の手順を実行します。

1. アプリケーションを生成したプロジェクトへのアクセス権限がない場合は、権限あるチームメイトから、アプリケーションのiOS App Store Package（IPA）ファイルを入手してください。あるいは次のように、自分で生成することもできます。
   a. Xcodeでプロジェクトを開きます。
   b. アプリケーションをアーカイブしてください。
   c. 次にアプリケーションのIPAファイルを生成します。
2. アプリケーションのユーザテスト用プロビジョニングプロファイルを、iOSプロビジョニングポータルから、自分のファイルシステムにダウンロードします（このプロファイルをXcodeにインストールする必要はありません）。

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

3. テスターに電子メールで、ユーザテスト用プロビジョニングプロファイルとIPAファイルを送ってください。

テスターからのクラッシュログのインポート

デバイスオーガナイザにテスターのクラッシュログを追加するには、クラッシュログを「Library」セクションの「Device Logs」グループへドラッグします。すると関連するシンボルに関する情報を表示できるようになります。

Important Xcodeでクラッシュログをシンボル化する（使用されたAPIに関する情報をクラッシュログに追加するため）には、Spotlightによって、アーカイブしたアプリケーションとそれらに対応するdSMYファイルが保存されているボリュームの索引付けを行う必要があります。

アプリケーションのテスター向けの説明書

このセクションは、テスターが各自のデバイス上でiOSアプリケーションをテストする手順についての、アプリケーションテスター向けの説明書です。自分のアプリケーションのテストに興味を持ってくれたユーザに、アプリケーションのテストに必要な特殊な作業の説明書と一緒に、これらの説明書を送信できます。

デベロッパへのIDの送信

テスト用アプリケーションをテスターに送信する前に、デベロッパはAppleのアプリケーションテストプログラムにテスターのデバイスを登録する必要があります。

デバイスIDをデベロッパに送信するには、次の手順を実行します。

1. iTunesを起動します。
2. デバイスをコンピュータに接続します。
3. 「デバイス(Devices)」リストからそのデバイスを選択します。
4. 「概要(Summary)」ペインで、「シリアル番号(Serial Number)」ラベルをクリックします。

   ラベルが「識別子(Identifier)」に変わります。
5. 「編集(Edit)」＞「コピー(Copy)」を選びます。
6. デバイス識別子をデベロッパに電子メールで送信します。

   電子メールには、忘れずに自分の名前を含めます。

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

テスト用アプリケーションのインストール

デベロッパからテスト用のアプリケーションが送られてきたら、iTunesを使ってデバイスにインストールしてください。

テストアプリケーションをデバイスにインストールするには、以下の手順に従います。

1. Finder上で、プロビジョニングプロファイル（拡張子が`.mobileprovision`のファイル）をDockのiTunesアイコンにドラッグします。
2. アプリケーションのアーカイブ（`<App_Name>.ipa`）をダブルクリックします。

   iTunesの「アプリケーション(Applications)」リストにアプリケーションが表示されます。
3. デバイスを同期します。

   デバイスのiOSのバージョンが、テストアプリケーションを実行できるバージョンより古い場合は、最新バージョンのiOSでデバイスを更新する必要があります。

デベロッパへのクラッシュレポートの送信

テスト中のアプリケーションがクラッシュした場合、iOSはそのイベントのレコードを作成します。次回デバイスをiTunesに接続したときに、iTunesはそれらのレコード（クラッシュログと呼ばれる）をコンピュータにダウンロードします。問題の解決を支援するために、テスト中のアプリケーションのクラッシュログをデベロッパに送信する必要があります。

Macからのクラッシュレポートの送信

クラッシュレポートをデベロッパに送信するには、次の手順を実行します。

1. Finderで、新しいウインドウを開きます。
2. 「移動」＞「フォルダへ移動」を選びます。
3. `~/Library/Logs/CrashReporter/MobileDevice`と入力します。
4. デバイス名で決まるフォルダを開きます。
5. テスト中のアプリケーションから始まる名前を持つクラッシュログを選択します。
6. 「Finder」＞「サービス」＞「Mail」＞「ファイルを送信」を選びます。
7. 「新規メッセージ」ウインドウで、「宛先」フィールドにデベロッパの電子メールアドレスを入力し、「件名」フィールドに「`<app_name> crash logs from <your_name>`」（たとえば、「`MyTestApp crash logs from Anna Haro`」）と入力します。
8. 「メッセージ」＞「送信」を選びます。
9. 同じレポートの重複送信を避けるため、送信済みのクラッシュログを削除しても構いません。

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

Windowsからのクラッシュレポートの送信

クラッシュレポートをデベロッパに送信するには、次の手順を実行します。

1. 使っているオペレーティングシステムに応じて、次のように、クラッシュログディレクトリを「Windows」検索フィールドに入力してください。ここで「<user_name>」の部分は、Windowsユーザ名で置き換えます。
   - Windows Vistaの場合のクラッシュログの格納場所

     ```
     C:\Users\<user_name>\AppData\Roaming\Apple
     computer\Logs\CrashReporter/MobileDevice
     ```

   - Windows XPの場合のクラッシュログの格納場所

     ```
     C:\Documents and Settings\<user_name>\Application Data\Apple
     computer\Logs\CrashReporter
     ```

2. デバイス名で始まる名前のフォルダを開き、テスト中のアプリケーションのクラッシュログをアプリケーションのデベロッパにメールで送信します。その際、件名の形式を<app_name> crash logs from <your_name> （たとえば、「MyTestApp crash logs from Anna Haro」）にします。

App Storeでのアプリケーションの公開

App Storeを通して一般配布用にアプリケーションを公開する準備が整ったら、iTunes Connectにアプリケーションを投稿します。このセクションでは、投稿用にアプリケーションを準備する方法、およびiTunes Connectにアプリケーションを投稿する方法について説明します。

Important App Store上でアプリケーションを公開できるのはチームのエージェントだけです。]

アプリケーションのDistributionプロビジョニングプロファイルの作成

アプリケーションのDistributionプロビジョニングプロファイルを作成するには、次の手順に従います。

1. 次の特性でiOSプロビジョニングポータルにDistributionプロファイルを作成します。

配布メソッド	App Store
プロファイル名	<App_Name> Distributionプロファイル

アプリケーションID	配布するアプリケーションの適切なアプリケーションID

プロビジョニングプロファイルの名前に、アプリケーション名を含めるか否かは任意です。

アプリケーションIDがあれば、個々のアプリケーション、あるいはアプリケーションドメインを識別できます。

2. Distributionプロビジョニングプロファイルをダウンロードしてください。
3. これをプロビジョニングプロファイルライブラリにインストールします。

   `<Profile_Name>.mobileprovision`ファイルを、デバイスオーガナイザの「Library」セクション以下、「Provisioning Profiles」リストにドラッグしてください。

アプリケーションをApp Storeで公開するよう登録

アプリケーションをApp Storeで公開するためには、iTunes Connectで、必要な情報を入力する必要があります。iTunes Connectについては、https://itunesconnect.apple.comを参照してください。

Important アプリケーションを検証し、あるいは公開するよう登録するためには、その操作をする人がiTunes Connectに登録されていなければなりません。

アプリケーションを検証し、あるいはApp Storeで公開するよう登録する手順を以下に示します。

1. AppStoreスキームを使ってアプリケーションのアーカイブを作成します。

   AppStoreスキームを設定する手順については、“App Storeに登録するためのプロジェクト設定” （22 ページ）を参照してください。
2. アプリケーションレコードがiTunes Connectにあることを確認します。このレコードの情報が、検証し、または公開するよう登録するアプリケーションと一致していなければなりません。

   特に次の項目を確認してください。

   - アプリケーション名
   - バンドルID
   - バージョン
3. iTunes Connectにおけるアプリケーションレコードの状態が、少なくとも「Waiting for Upload」であることを確認してください。
4. 「配布用にアプリケーションを登録」。

   iTunes Connect検証テストに合格しなければ、アプリケーションを登録し、配布することはできません。

トラブルシューティング「Unable to find registered user with username <username>」

「Unable to find registered user with username <username>」というダイアログが現れた場合、iTunes Connectに登録されていないことを表します。

- 登録するよう、チームの権限がある人に依頼してください。

iOS開発：トラブルシューティング

この章では、アプリケーションを開発し、App Storeに登録、公開する際に起こりうる、さまざまな障害への対処方法を解説します。

このドキュメントの対象　以下の内容は、Xcode 4.3およびiOS SDK 5.0を対象としています。

問題

証明書に関する問題

開発用証明書が期限切れである

期限切れの開発用証明書は、次の手順で更新してください。

- 新しい開発用証明書を作成するようチームに要求します。

  具体的な手順は"期限切れになった証明書の更新"（15 ページ）を参照してください。

開発用証明書が無効である

証明書が無効なのは、対応する秘密鍵がキーチェーンにないためかも知れません。

有効な証明書に更新する場合は、無効になった証明書をキーチェーンから削除した上で、次のいずれかの操作をしてください。

- 証明書と対応する秘密鍵を、署名アセットのバックアップから復元する。

  具体的な手順は"署名/プロビジョニングアセットの保全と転送"（16 ページ）を参照してください。

- コード署名アセットをリセットする。

  具体的な手順は"自分および開発チームのMac上にある、署名アセット、プロビジョニングアセットをリセットする"（57 ページ）を参照してください。

開発用証明書がキーチェーンに登録されていない

証明書がキーチェーンにないのは、現在開発に使っているMacが、今まで使っていたものと違うからかも知れません。

> **Important** 開発用証明書をキーチェーンに追加するためには、開発チームに属してる必要があります。

キーチェーンに開発用証明書を追加するには、次のいずれかの操作をしてください。

- 署名アセットを、以前開発に使っていたMacから転送する。

  具体的な手順は"署名/プロビジョニングアセットの保全と転送"（16 ページ）を参照してください。

- コード署名アセットをリセットする。

  具体的な手順は"自分および開発チームのMac上にある、署名アセット、プロビジョニングアセットをリセットする"（57 ページ）を参照してください。

Apple Worldwide Developer Relations Certification Authorityの証明書がキーチェーンにない

キーチェーンの開発用証明書や配布用証明書は、Apple Worldwide Developer Relations Certification Authorityの証明書がなければ有効になりません。

> **Important** Apple Worldwide Developer Relations Certification Authorityの証明書が開発チームの署名アセットにない詳細については"開発チームへの加入"（9 ページ）を参照してください。

この証明書は次のようにしてキーチェーンに追加します。

1. チームの適切な証明書を使って、iOSプロビジョニングポータルにログインします。
2. Apple Worldwide Developer Relations Certification Authorityの証明書をダウンロードしてください。
3. 証明書をダブルクリックするか、Dockの「キーチェーンアクセス」アイコン上にドラッグします。

プロビジョニングに関する問題

Xcodeがアプリケーションを開発用デバイスにインストールできない

Xcodeがアプリケーションを開発用デバイスにインストールできない場合、当該アプリケーション用のプロビジョニングプロファイルに問題があるかも知れません。

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

- プロビジョニングプロファイルが、開発チームの署名アセットに適切に設定されたものであることを確認してください。

  プロビジョニングプロファイルの設定については、“開発用デバイスの設定”（10 ページ）を参照してください。

プロビジョニングプロファイルが期限切れになった

開発用デバイスに格納されているプロビジョニングプロファイルが期限切れになると、Xcodeは当該デバイスにアプリケーションをインストールできません。

次のように対処してください。

- 期限切れになったプロファイルを、新しいコピーに置き換えます。

  具体的な手順は“もうすぐ期限切れになる、またはすでに期限切れになったプロビジョニングプロファイルの更新”（16 ページ）を参照してください。

ビルド作業に関する問題

コード署名エラー

この節では、ビルドに失敗する原因である、コード署名IDの問題とその対処法を解説します。

Xcodeがプロビジョニングプロファイルを見つけられない

プロビジョニングプロファイルを取得し更新した後、そのアプリケーションIDが変わったなどのため、このエラーメッセージが現れるようになる場合があります。

```
Code Sign error: Provisioning Profile 'xxxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx' can't
 be found
```

次のように対処してください。

1. 「Code Signing Identity」ビルド設定の値が、正しいプロビジョニングプロファイルとコード署名IDであることを確認します。

   “コード署名IDの設定”（26 ページ）を参照してください。

2. キーチェーンの署名IDが正しいことを確認します。

   “開発用署名IDがキーチェーンに登録されていることを確認する”（55 ページ）を参照してください。

Xcodeが証明書を信頼しない

開発用証明書や配布用証明書の正当性をXcodeが確認できない場合に、このエラーメッセージが現れます。

```
Code Sign error: CSSMERR_TP_NOT_TRUSTED
```

次のように対処してください。

- 証明書の信頼レベルが正しいことを確認します。

  具体的な手順は、“証明書の信頼レベルを正しくする”（56 ページ）を参照してください。

コード署名IDのビルド設定が、キーチェーンの有効なコード署名IDを識別できない。

証明書が期限切れになったなど、無効である場合にこのエラーメッセージが現れます。

```
Code Signing Identity 'iPhone Developer' does not match any valid, non-expired,
code-signing certificate in your keychain.
```

次のように対処してください。

1. 証明書が登録されたプロビジョニングプロファイルをダウンロードしてください。
   a. 「Window」＞「Organizer」を実行し、開いた画面で「Devices」をクリックします。
   b. 「ライブラリ(Library)」セクションで、「プロビジョニングプロファイル(Provisioning Profiles)」を選択します。
   c. 「Refresh」をクリックしてください。
2. “コード署名IDの設定”（26 ページ）の説明に従って、有効なコード署名IDを選びます。

キーチェーンにコード署名IDが重複して登録されている

キーチェーンにコード署名IDが重複していると、開発用IDが重複している、配布用IDが重複している、などのエラーメッセージが現れます（キーチェーンにはそれぞれのIDを1つずつしか登録できません）。

```
Build error "iPhone Developer: <your_name> (XYZ123ABC): ambiguous (matches "iPhone
 Developer: <your_name> (XYZ123ABC)" in /Library/Keychains/System.keychain and
"iPhone Developer: <your_name> (XYZ123ABC)" in
/Users/../Library/Keychains/login.keychain)"
```

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

```
[BEROR]CodeSign error: Certificate identity 'iPhone Distribution: <your_name>'
appears more than once in the keychain. The codesign tool requires there only be
one.
```

対処法として、次の方法を試してみてください。

- キーチェーンの「My Certificates」リストから重複しているIDを削除する。
- コード署名アセットをリセットする。

  具体的な手順は“自分および開発チームのMac上にある、署名アセット、プロビジョニングアセットをリセットする”（57 ページ）を参照してください。

アプリケーションIDのエラー

プロビジョニングプロファイルのアプリケーションIDが、アプリケーションのバンドルIDと一致しない

「コード署名ID(Code Signing Identity)」ビルド設定で選択されたプロビジョニングプロファイルのアプリケーションIDと、アプリケーションバンドルのIDが矛盾する場合、次のようなエラーメッセージが現れます。

```
Code Sign error: Provisioning profile 'MyApp Profile' specifies the Application
Identifier 'com.mycompany.MyApp.*' which doesn't match the current setting
'com.mycompany.MyApp'
```

この場合、次の点を確認してください。

- アプリケーションのバンドルIDが正しく設定されていること。
- 「コード署名ID(Code Signing Identity)」ビルド設定で指定されたプロビジョニングプロファイルが正しいこと。
- プロビジョニングプロファイルが使っているアプリケーションIDが正しいこと。

  アプリケーションIDの設定方法については、“Creating and Configuring App IDs”を参照してください。

実行デスティネーションとしてデバイスが表示されない

プロジェクトやワークスペースを開いたとき、「Scheme」ツールバーメニューに、実行デスティネーションとして、接続したデバイスが表示されない場合は、次のように対処してください。

1. アプリケーションが動作するiOSバージョンが、デバイスにインストールされたiOSバージョンを含んでいることを確認します。詳しくは“アプリケーションが動作するiOSバージョンの指定”（17 ページ）および“デバイスにiOSをインストール”（13 ページ）を参照してください。

2. デバイス上に有効なプロビジョニングプロファイルがあることを確認します。“開発用デバイスの設定”（10 ページ）を参照してください。
3. プロジェクトで使っているiOS SDKのバージョン番号が、当該デバイスのiOSバージョンのバージョン以降であることを確かめます。

   たとえば、Xcodeには「iOS SDK 4.3」と表示されるけれども、デバイスにはiOS 5.0がインストールされている場合、iOS SDK 5.0が付属するXcodeバージョンをMacにインストールしなければなりません。

デバッグ情報に関する問題

デバイスを接続したとき、Xcodeに「Unknown iOS Detected」というダイアログが表示される

デバイス上でアプリケーションをデバッグするためには、当該デバイスに関する情報を、Xcodeが取得できなければなりません。

問題の解決手順

開発用署名IDがキーチェーンに登録されていることを確認する

アプリケーションをビルドし、アーカイブし、あるいは開発デバイスにインストールする際、Xcodeに、コード署名に関するエラーが表示されることがあります。開発用署名ID（開発用証明書と秘密鍵）が見つからない、あるいはプロジェクトで使っている開発用証明書が無効である、などの場合です。

開発用署名IDがキーチェーンに登録されているか否かは、次の手順で確認してください。

1. キーチェーンアクセスを起動します。
2. 「Category」リストの「My Certificates」をクリックします。
3. ウインドウ右上隅の検索フィールドに、「`iphone developer`」と入力してください。
4. 証明書リストに「iPhone Developer: <Your_Name>」という証明書があることを確認します。

   ない場合は、“開発用証明書がキーチェーンに登録されていない”（51 ページ）を参照してください。
5. 証明書の秘密鍵を表示します。証明書の左側にある三角形をクリックしてください。
6. 証明書を選択し、有効である旨が表示されることを確認します。

   無効である場合は、“開発用証明書が無効である”（50 ページ）を参照してください。

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

これまでiOS開発に使ったことがないMac上で新たにアプリケーション開発を始める場合、今まで使っていたMacから署名アセットを転送する必要があるかも知れません。具体的な手順は"署名/プロビジョニングアセットの保全と転送"（16 ページ）を参照してください。

証明書の信頼レベルを正しくする

証明書の信頼レベルをシステムのデフォルト値に戻す手順を以下に示します。

1. キーチェーンアクセスを起動します。
2. 「Category」リストから「My Certificates」を選択し、検索フィールドに「iphone」と入力してください。
3. 証明書リストのiPhone証明書それぞれについて、次のように操作します。
   a. 証明書をダブルクリックしてください。

      「This certificate was signed by an unknown authority」というメッセージが現れた場合、"Apple Worldwide Developer Relations Certification Authorityの証明書がキーチェーンにない"（51 ページ）を参照してください。
   b. 証明書ウインドウの該当する三角形をクリックして、「信頼(Trust)」セクションを表示します。
   c. 「この証明書を使用するとき(When using this certificate)」として、「システムデフォルトを使用(Use System Defaults)」を選択します。
   d. 証明書ウインドウを閉じてください。
   e. 証明書情報を調べ、有効である旨の表示があることを確認します。

      無効である場合は、"開発用証明書が無効である"（50 ページ）を参照してください。
4. 「分類(Category)」リストから「証明書(Certificates)」を選択し、検索フィールドに「apple worldwide」と入力してください。

   「Apple Worldwide Developer Relations Certificate Authority」という証明書が見つからない場合は"Apple Worldwide Developer Relations Certification Authorityの証明書がキーチェーンにない"（51 ページ）に従って対処してください。
5. 証明書リストに並んでいる「Apple Worldwide Developer Authority」証明書それぞれについて、次のように操作します。
   a. 証明書をダブルクリックしてください。
   b. 証明書ウインドウの該当する三角形をクリックして、「信頼(Trust)」セクションを表示します。
   c. 「この証明書を使用するとき(When using this certificate)」として、「システムデフォルトを使用(Use System Defaults)」を選択します。

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

d. 証明書ウインドウを閉じてください。

e. 証明書情報を調べ、有効である旨の表示があることを確認します。

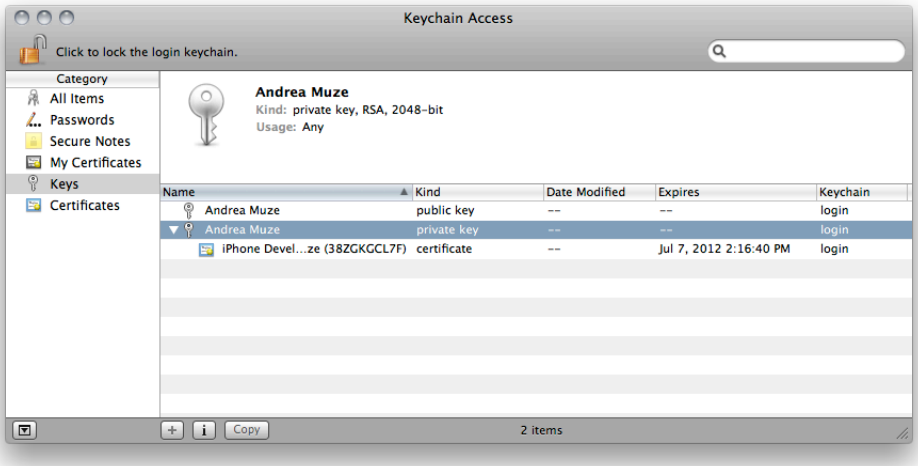
無効である場合は、“開発用証明書が無効である”（50 ページ）を参照してください。

自分および開発チームのMac上にある、署名アセット、プロビジョニングアセットをリセットする

署名アセット、プロビジョニングアセットのバックアップがない場合（“署名/プロビジョニングアセットの保全と転送”（16 ページ））、自分および開発チームのMac上で生成し直さなければなりません。

署名アセット、プロビジョニングアセットをリセットする手順を以下に示します。

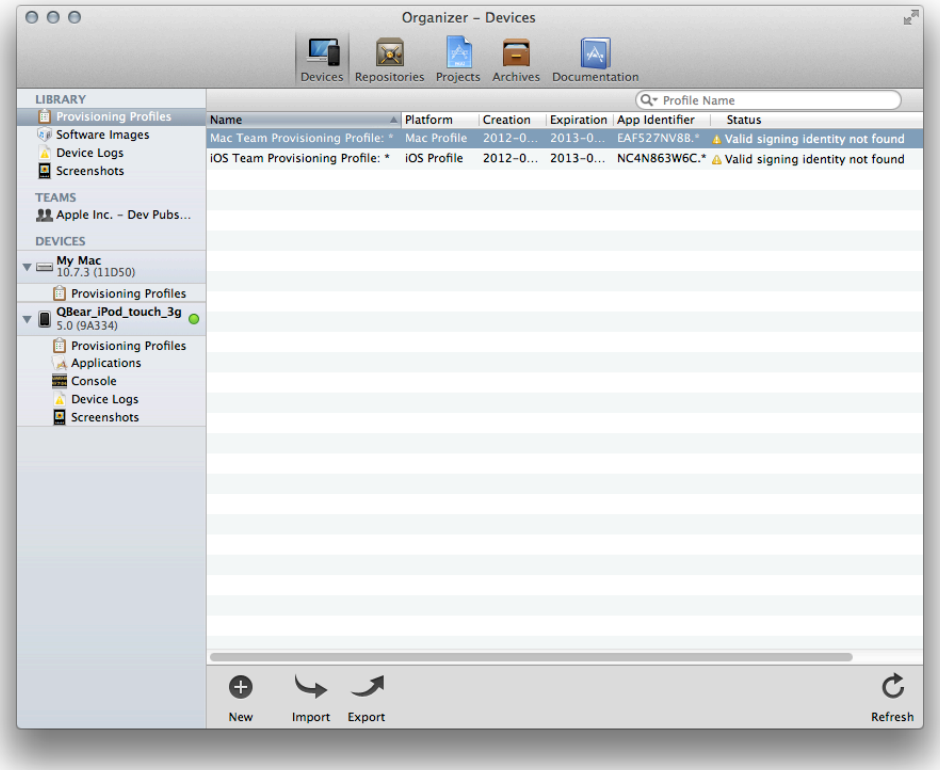
1. Xcodeを終了してください。
2. 秘密鍵、公開鍵、証明書を、次の手順でキーチェーンから削除します。
   a. 「キーチェーンアクセス」（`/Applications/Utilities`以下）を起動してください。
   b. 「分類(Category)」セクションで「鍵(Keys)」を選択します。
   c. 隣に三角形がついている秘密鍵それぞれについて、次のように操作してください。
      a. 三角形をクリックして、秘密鍵に関する情報を表示します。



      b. 開発者証明書または配布用証明書に対応する秘密鍵であれば、この鍵と、対応する公開鍵を選択し、Deleteキーを押してください。
   d. 「分類(Category)」セクションで「自分の証明書(My Certificates)」を選択し、残っている開発者証明書、配布用証明書を削除します。
3. チームの署名アセットの該当する証明書を無効にします。
   a. チームの適切な証明書を使って、iOSプロビジョニングポータルにログインします。
   b. 開発用/配布用証明書を無効にします。

4. デバイスオーガナイザに証明書が表示されないことを、次の手順で確認してください。
   a. Xcodeを起動します。
   b. 「ウインドウ(Window)」＞「オーガナイザ(Organizer)」コマンドで「オーガナイザ(Organizer)」ウインドウを開き、「Devices」をクリックしてデバイスオーガナイザを開きます。
   c. 「Library」セクションで、「Developer Profile」を選択します。
   d. 証明書リスト（「Developer Profile」ペインの一番上）に証明書がないことを確認してください。
5. プロビジョニングプロファイルがすべて無効であることを、次の手順で確認します。

   デバイスオーガナイザの「Library」セクションで、「Provisioning Profiles」を選択してください。



   有効なプロビジョニングプロファイルが残っている場合は、チームの署名アセットで、証明書を確実に無効にしてください。
6. プロビジョニングプロファイルを、デバイスオーガナイザおよび開発デバイスから、次の手順で削除します。
   a. デバイスオーガナイザの「Library」セクションで、「Provisioning Profiles」を選択してください。
   b. プロビジョニングプロファイルをすべて選択し、Deleteキーを押してください。
   c. 開発デバイスそれぞれについて、次のように操作します。

a. デバイスを接続します。

b. デバイスのグループで「Provisioning Profiles」を選択します。

c. 無効なプロファイルを選択し、Deleteキーを押してください。

7. 新しい証明書を、次の手順で生成します。

a. デバイスオーガナイザの「Library」セクションで、「Provisioning Profiles」を選択してください。

b. 「Refresh」をクリックしてください。

Xcodeが開発者証明書要求を作成します。これに基づき、開発者証明書を要求するようXcodeに指示してください。

チームのエージェントであれば、配布用証明書要求も作成されます。これに基づき、配布用証明書を要求するようXcodeに指示してください。

しばらくすると、チームプロビジョニングプロファイルがリストに現れます。

8. チームの署名アセットで、新しい証明書に、適切なプロビジョニングプロファイルを対応づけてください。

チームの管理者やエージェントであれば、各プロビジョニングプロファイルに、自分の証明書を追加します。そうでなければ、管理者またはエージェントに、追加するよう依頼してください。

9. 新しいプロビジョニングプロファイルを、デバイスオーガナイザおよびデバイスに、次の手順でインストールします。

a. デバイスオーガナイザの「Library」セクションで、「Provisioning Profiles」を選択してください。

b. 「Refresh」をクリックしてください。

開発者証明書に対応するプロビジョニングプロファイルが、リストに表示されます。

> **Important** チームのエージェントであれば、XcodeはDistributionプロファイルをダウンロードしないことに注意してください。アプリケーションを配布する際は、別途ダウンロードする必要があります。詳細については、“アプリケーションの配布”（42 ページ）を参照してください。

iOS開発：FAQ

以下は、iOSの開発についてデベロッパから寄せられるよくある質問です。

- デバイス上でアプリケーションを実行するには?

  "実行デスティネーションの指定"（28 ページ）を参照してください。

- Core Locationフレームワークをプロジェクトに追加するには?

  プロジェクトエディタで、Core Locationフレームワークにリンクします。

- iOSシミュレータアプリケーションは、ネットワークホームディレクトリで動作しますか。

  いいえ。

- Objective-C言語のプロパティを利用できるようにするには、インスタンス変数またはアクセサメソッドで支える必要がありますか。

  はい。

- スタティックライブラリは、iOSアプリケーションで使用する前にコード署名する必要がありますか。

  いいえ。

- なぜアプリケーションでPNGファイルの処理に問題が発生するのですか。

  PNGファイルを使おうとしているコードがPNGの圧縮ファイルを理解できていない可能性があります。

  「Compress PNG Files」ビルド設定をオフにしてください。

- WindowsでiOSアプリケーションを開発できますか。

  いいえ。iOSアプリケーションを開発できるのは、Mac OS Xのみです。

- スタティックライブラリのすべてのObjective-Cクラスをどのようにリンクすれは良いですか。

  「ほかのリンカフラグ(Other Linker Flags)」ビルド設定を`-ObjC`に設定します。このようにしてもすべてのクラスをリンクしない場合は、`-all_load`に設定します。

- 廃止されたAPIはいつ置き換えるべきですか。

  アプリケーションを実行させるiOSバージョンを検討し、できる限り迅速にアップデートします。詳細については、『*SDK Compatibility Guide*』を参照してください。

- iOSシミュレータでコンピュータのカメラを使用できますか。

  いいえ。

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

- iOSの開発に必要な最低限のハードウェア要件は何ですか。

  Intelプロセッサを搭載したMacです。

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

スタティックライブラリを開発してアプリケーションに組み込む手順

アプリケーションに組み込むスタティックライブラリを開発する必要があり、製品ごとに独立したプロジェクトを用意しなければならない場合は、スタティックライブラリとアプリケーションのプロジェクトを包括的に扱える、ワークスペースを利用するとよいでしょう。この場合、ワークスペースにおけるプロジェクトの設定は、次のようにしてください。

このドキュメントの対象　以下の内容は、Xcode 4.3およびiOS SDK 5.0を対象としています。

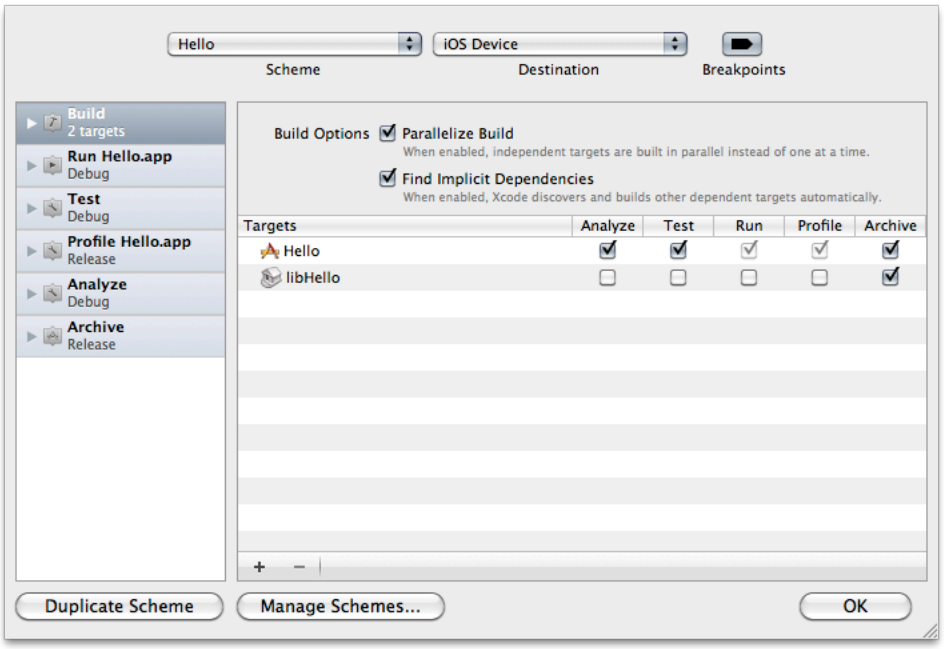
1. スタティックライブラリのビルド対象について：
   - エクスポートするヘッダは、「Copy Headers」ビルドフェーズの「Project」グループに置きます。
   - 「Skip Install」ビルド設定を「Yes」とします。
2. アプリケーションのビルド対象について：
   - 「User Header Search Paths」ビルド設定として、スタティックライブラリのヘッダファイルを置いたディレクトリの絶対パスを指定します。

     Important スタティックライブラリのプロジェクトディレクトリを、同じファイルシステム上の別の場所に移動する場合は、「User Header Search Paths」ビルド設定の値も、ヘッダファイルの新しい位置に合わせて修正する必要があります。

   - 「Always Search User Paths」ビルド設定を「Yes」とします。
   - 「Skip Install」ビルド設定を「No」とします。

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

3. アプリケーションをビルドするスキームで、スタティックライブラリも併せてビルドし、アーカイブするようになっていることを確認してください。



用語解説

アプリケーションID（Application ID） 1つのベンダのiOSアプリケーションまたはiOSアプリケーションセットを識別する文字列。これは、バンドル識別子に類似しています。アプリケーションIDの例を以下に示します。
`GFWOTNXFIY.com.mycompany.MyApp`,
`GFWOTNXFIY.com.mycompany.*`.

ベースSDK（Base SDK） プロジェクトのターゲットをビルドするときに使用するデフォルトのSDKを指定するプロジェクト設定。個々のターゲットでは、これと異なる設定ができます。

ビルド構成（build configuration） プロダクトをビルドするプロジェクトごとに、名前をつけた一連のビルド設定をまとめたもの。デバッグ用、リリース用など、目的に応じて使い分けます。

CSR（証明書署名要求：Certificate Signing Request） 開発用証明書を生成するために使われる個人情報を含むファイル。

コード補完（code completion） 識別子やキーワードが入力されるたびに、考えられる完成形を自動的に提案するショートカット機能。提案は、入力されたテキストとその周辺の文脈に基づいて行われます。

開発者証明書（Developer Certificate） iOSアプリケーションデベロッパを識別するファイル。Xcodeは、開発者証明書を使用して、アプリケーションバイナリに署名します。

開発チーム（development team） 法的主体（個人、会社、組織）、個人（チームのアプリケーションの開発や配布に関与する人）が連携し、オンライン開発リソース（iOS/Macプロビジョニングポータル）を利用することにより、チームの人員が開発デバイス上でアプリケーションをビルド、実行し、顧客に配布するための仕組み。

デバイスタイプ（Device Type） iOSが稼働できるデバイスのタイプ。iPhone（iPod touchを含む）とiPadの2種類があります。

Entitlement アプリケーションが、保護されたiOS機能などにアクセスできるようにするプロパティ。

GPS eXhange Format（GPX） 単一または一連のロケーション（中間地点）を表すためのファイル形式。

instrument Instrumentsアプリケーションを使用して開発されたデータ収集エージェント。Instrumentsは、1つのアプリケーションまたはシステム全体に関するパフォーマンス情報を収集します。

Instrumentsアプリケーション（Instruments application） アプリケーションのパフォーマンスデータを収集したり発掘したりする、グラフィック表示のパフォーマンス分析ツール。

iOS Dev Center iOSアプリケーションの開発に必要なすべてのリソースを提供するAppleのデベロッパセンター。アクセスするには、登録アップルデベロッパになる必要があります。

iOSデベロッパプログラム（iOS Developer Program） iOSアプリケーションの開発、iOSベースのデバイス上でのテスト、ユーザへの配布を可能にするプログラム。

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

iOSプロビジョニングポータル（iOS Provisioning Portal） iOSアプリケーションをテストするためのデバイスを構成できるようにする、iOS Dev Centerのアクセス制限領域。

iOSシミュレータアプリケーション（iOS Simulator application） 開発の初期段階でiOSアプリケーションをテストするため、iOSベースのデバイスの動作をシミュレートする、Xcodeに付属のアプリケーション。詳細については、『**シミュレータ**』を参照してください。

プロビジョニングプロファイル（Provisioning Profile） 開発中のアプリケーションをiOSベースのデバイスにインストールできるようにするファイル。このファイルには、1つ以上の開発用証明書、1つのアプリケーションID、および1つ以上のデバイスIDが含まれています。

シミュレーション環境（simulation environment） iOSシミュレータアプリケーションが、iOSリリースの動作をシミュレートするために用いる、表面には現れない環境。シミュレーション環境ごとに独自のシステムファイルがあり、アプリケーションをインストールできます。iOSシミュレータは、iOSのバージョンごとにシミュレーション環境を提供します。

スキーム（scheme） 製品を実行し、テストし、分析し、アーカイブする手順をまとめたもの。

シミュレータ（simulator） あるiOSリリースが稼働するiOSデバイスをシミュレートするソフトウェア。iOSシミュレータアプリケーションのシミュレーション環境で動作します。『**iOSシミュレータアプリケーション**』、『**シミュレーション環境**』も参照してください。

スタティックライブラリ（static library） アプリケーションのバイナリファイルに、リンク時に組み込まれるオブジェクトコード。

テストケース（test case） テスト対象コードを実行してそれが期待通りに動作することを確認するためのコード。テストケースは**テストケースメソッド**で実装されます。

テストケースメソッド（test-case method） `test...`という名前の単体テストクラスのインスタンス。テストにAPIを呼び出し、期待する結果が得られるかレポートします。

ユーザテスト用プロビジョニングプロファイル（user testing provisioning profile） 開発チームに属していないユーザに発行されるプロビジョニングプロファイル。これを利用すると、チームがまだApp Storeに公開していないアプリケーションのインストールとテストができます。**暫定プロビジョニングプロファイル**とも呼ばれます。

テストスイート（test suite） テストケースセット。**テストケース（test case）**も参照。

Xcode Cocoaアプリケーション（Mac OS X用）およびCocoa Touchアプリケーション（iOS用）の開発に使われる一連のツールとリソース。

Xcodeアプリケーション（Xcode application） Xcode統合開発環境（IDE）のメインアプリケーション。Xcode IDEに含まれるほかのアプリケーションを管理し、ソフトウェア製品の開発に使われるメインのユーザインターフェイスを提供します。

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

書類の改訂履歴

この表は「*iOSツールワークフローガイド*」の改訂履歴です。

日付	メモ
2012-03-01	Xcode 4.3のワークフローを取り込みました。
2012-01-09	ロケーションのシミュレーション、アプリケーションデータの管理、配布専用アセットの取得に関する情報を追加しました。 ロケーションやトラックを、シミュレータ上やデバイス上でシミュレートすることに関する情報を、“ロケーションまたはトラックの指定” （29 ページ）に追加しました。 “アプリケーションデータの管理” （30 ページ）にワークフローを追加して更新し、起動時にアプリケーションデータを指定する方法について“アプリケーションデータの指定” （28 ページ）に説明しました。 配布専用アセットの取得に関する情報を“配布専用アセットの準備” （15 ページ）に追加しました。 単体テストに関する情報の大部分を『*Xcode Unit Testing Guide*』に移動しました。 『*iOS App Development Workflow Guide*』からドキュメント名を変更しました。 iCloudエンタイトルメントについての情報を追加し、トラブルシューティングの章に加筆しました。 “iCloudエンタイトルメントの設定” （17 ページ）を追加しました。 “App Storeに登録するためのプロジェクト設定” （22 ページ）を追加しました。 “iOS開発：トラブルシューティング” （50 ページ）を追加しました。

© 2012 Apple Inc. All Rights Reserved.

日付	メモ
	“アプリケーションをApp Storeで公開するよう登録”（48 ページ）の記述を修正しました。
2011-09-09	Xcode 4.0のユーザインターフェイスおよび概念を追加しました。 “スタティックライブラリを開発してアプリケーションに組み込む手順”（62 ページ）を追加しました。 「プロジェクトの単体テスト設定をXcode 3.2.5からXcode 4.0.2に移行」を追加しました。
2010-11-15	ベースSDKのビルド設定への変更についてドキュメント化しました。 “アプリケーションのビルドに用いるSDKの設定”（25 ページ）に、「ベースSDK(Base SDK)」の「最新のiOS(Latest iOS)」の値についての説明と、「Base SDK Missing（ベースSDKがない）」というエラーの解決方法の手順を追加しました。 “ジェスチャの実行”（34 ページ）に、iOSシミュレータ上でピンチの中心を移動する手順を追加しました。 iOS SDK 4.2配布版のワークフローと必要条件に関する内容を更新しました。
2010-08-26	細かな訂正を行いました。
2010-07-02	『iPhone開発ガイド』からドキュメント名を変更しました。「Hello, World!」チュートリアルをiPhone SDK 4.0用に更新しました。 「Hello, World!」のチュートリアルとソースコード（「チュートリアル：Hello, World!」と「Hello, World!ソースコード」）を、iOS SDK 4.0ツールセット用に更新しました。
2010-05-28	プロビジョニングプロファイルの自動管理、アプリケーションアーカイブ、およびアプリケーションの配布についての情報を追加しました。

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

日付	メモ
	“アプリケーションのビルドと実行”（23 ページ）を更新し、ベースSDK(Base SDK)と、「iPhone OS Deployment Target」ビルド設定、およびプロジェクトウインドウの「概要(Overview)」ツールバーメニューの使い方の詳細を記述しました。 “iOSシミュレータの使用”（33 ページ）を更新し、iOS 4.0のObjective-Cランタイムの変更が既存のiOSシミュレータバイナリに及ぼす影響についての情報を追加しました。 “開発アセット、配布アセットの設定”（9 ページ）を更新し、プロビジョニングプロファイル自動管理と、Xcodeの「オーガナイザ(Organizer)」での開発者プロファイルの管理方法について記述しました。 “アプリケーションの配布”（42 ページ）を更新し、「ビルドとアーカイブ(Build and Archive)」コマンドおよびテスト用アプリケーションの配布のための推奨されるワークフローについて詳細を追加しました。 “iOSシミュレータの使用”（33 ページ）にハードウェアシミュレーションのサポート情報を追加しました。
2010-03-19	iPadの情報を追加しました。 アプリケーションを実行するデバイスファミリを指定する方法を説明する“アプリケーションが動作するデバイスの指定”（18 ページ）を追加しました。 iPhoneターゲットをiPadアプリケーションのビルド用にアップグレードする方法を説明する“アプリケーションが動作するデバイスの指定”（18 ページ）を追加しました。 “iOSシミュレータの使用”（33 ページ）を更新し、iPadの情報を追加しました。
2010-01-20	誤字の訂正や、読者からのフィードバックの反映を行いました。
2009-08-06	プロパティリストファイルの編集、スタティックライブラリのリンク、およびiOSシミュレータのバージョンについての情報を追加しました。細かな変更を行いました。

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

日付	メモ
	“アプリケーションの設定”（17ページ）に「プロパティリストファイルの編集」を追加しました。 デバイス上でのアプリケーションのデバッグについての重要な情報を“デバッグ機能の概要”（38 ページ）に追加しました。 “デバイスとiOSバージョンの設定”（33 ページ）を追加しました。
2009-05-28	アプリケーションをビルドするアーキテクチャの設定方法について説明しました。 “アプリケーションが動作するアーキテクチャの指定”（19ページ）を追加し、アプリケーションのビルドが可能なアーキテクチャの選択方法を説明しました。
2009-05-14	単体テストの説明と効率化されたビルドワークフローを追加しました。 “高い品質と性能の保証”（37 ページ）を追加しました。 「iOSシミュレータのフレームワークとライブラリ」を追加しました。 “アプリケーションのビルドと実行”（23 ページ）を簡単なビルドワークフローを使って更新しました。バックアップからのアプリケーションデータの復元についての情報を追加しました。
2009-04-08	Entitlementプロパティリストファイルの作成についての情報を追加しました。内容を若干再編成しました。 「アプリケーションのエンタイトルメントの管理」を追加しました。
2009-03-04	細かな内容の変更を行いました。
2009-02-04	細かな内容の変更を行いました。

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

日付	メモ
	“スクリーンショットのキャプチャ”（14 ページ）を更新し、キャプチャしたスクリーンショットのPNGファイルをオーガナイザ（Organizer）に取り込む方法を示しました。 “アプリケーションのビルド”（29 ページ）をアプリケーションIDのビルドエラ情報で更新しました。
2009-01-06	細かな内容の追加を行いました。 iOSシミュレータバイナリは、シミュレータの1つのリリース上でのみ使用できるという説明を追加しました。
2008-11-14	iOSシミュレータの新しい機能についての情報を追加しました。 “ユーザテストの要件”（43 ページ）を追加しました。 “ハードウェアの操作”（34 ページ）に「メモリ警告をシミュレート(Simulate Memory Warning)」および「着信ステータスバーを切り替える(Toggle In-Call Status Bar)」のコマンドについての情報を追加しました。 「Core Location機能」を追加しました。 iOSアプリケーションでのスタティックライブラリの使用についての情報を、「iOSアプリケーションプロジェクトの作成」に追加しました。
2008-10-15	Xcodeを使ったiPhoneアプリケーションの開発方法について説明した新規ドキュメント。 『*iPhone OS Programming Guide*』および『*iOS Simulator Programming Guide*』に以前公開された内容を組み込みました。

2012-03-01 | © 2012 Apple Inc. All Rights Reserved.

Apple Inc.
© 2012 Apple Inc.
All rights reserved.

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写複製（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/

App Store is a service mark of Apple Inc.

iCloud is a registered service mark of Apple Inc.

Apple, the Apple logo, Cocoa, Cocoa Touch, FaceTime, Finder, Instruments, iPad, iPhone, iPhoto, iPod, iPod touch, iTunes, Keychain, Mac, Mac OS, Objective-C, OS X, Shake, Spotlight, and Xcode are trademarks of Apple Inc., registered in the United States and other countries.

Retina is a trademark of Apple Inc.

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

Intel and Intel Core are registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

OpenGL is a registered trademark of Silicon Graphics, Inc.

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します。

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。