

# Packager for iPhone® による ADOBE® AIR® アプリケーションの構築

## 法律上の注意

法律上の注意については、[http://help.adobe.com/ja\\_JP/legalnotices/index.html](http://help.adobe.com/ja_JP/legalnotices/index.html)を参照してください。

# 目次

## 第 1 章：iPhone 向け AIR アプリケーションの構築の開始

重要な概念 .....	1
アドビから開発ツールを入手 .....	4
Apple から開発ファイル入手 .....	4
Flash Professional CS5 での Hello World iPhone アプリケーションの作成 .....	8

## 第 2 章：iPhone アプリケーションのコンパイルとデバッグ

iPhone のアイコンと起動画面イメージ .....	13
iPhone アプリケーション設定 .....	15
iPhone アプリケーションインストーラー (IPA) ファイルのコンパイル .....	21
iPhone アプリケーションのインストール .....	23
iPhone アプリケーションのデバッグ .....	24
App Store への iPhone アプリケーションの送信 .....	27

## 第 3 章：ActionScript 3.0 API によるモバイルデバイスのサポート

モバイルデバイスでサポートされていない ActionScript 3.0 API .....	28
モバイル AIR アプリケーション固有の ActionScript API .....	30
モバイルアプリケーションの開発者に特に有用な ActionScript 3.0 API .....	34

## 第 4 章：iPhone アプリケーションのデザイン上の考慮事項

ハードウェアアクセラレーション .....	35
表示オブジェクトのパフォーマンスを向上させるその他の方法 .....	37
情報密度 .....	38
フォントとテキスト入力 .....	38
アプリケーションの状態の保存 .....	39
画面の向きの変更 .....	40
ヒットターゲット .....	40
メモリの割り当て .....	40
描画 API .....	40
イベントバブリング .....	40
ビデオパフォーマンスの最適化 .....	41
Flex と Flash のコンポーネント .....	41
アプリケーションのファイルサイズの抑制 .....	41

# 第 1 章: iPhone 向け AIR アプリケーションの構築の開始

Adobe® Flash® Platform ツールおよび ActionScript® 3.0 コードを使用して、iPhone および iPod Touch 用の Adobe® AIR® アプリケーションを構築できます。これらのアプリケーションは、他の iPhone アプリケーションと同様の方法で、配布、インストールおよび実行できます。

**注意:** このドキュメントでは、iPhone と iPod Touch をまとめて「iPhone」と呼びます。

Adobe® Flash® Professional CS5 には Packager for iPhone® が付属しています。Packager for iPhone は、ActionScript 3.0 バイトコードを iPhone のネイティブアプリケーションコードにコンパイルします。iPhone アプリケーションは、iPhone アプリケーションインストーラーファイル (.ipa ファイル) として、iTunes Store 経由で配布されます。

Flash Professional CS5 または Adobe® Flash® Builder™ 4 を使用すると、アプリケーションのソースの ActionScript 3.0 コンテンツを編集できます。

iPhone アプリケーションを開発するには、Flash Professional CS5 を使用します。

また、iPhone 開発用証明書を Apple から入手する必要があります。

**重要:** iPhone アプリケーションを開発する前に、iPhone 用アプリケーションのデザインに関する情報を確認してください。35 ページの「[iPhone アプリケーションのデザイン上の考慮事項](#)」を参照してください。また、iPhone アプリケーション構築に必要な開発ファイルについての知識も必要です。4 ページの「[Apple から開発ファイル入手](#)」を参照してください。

## 重要な概念

ActionScript 3.0 を使用して iPhone アプリケーションを開発する前に、関連する概念とワークフローについて理解することが重要です。

## 用語集

iPhone アプリケーションの構築にあたっては、次の各用語を理解しておくことが重要です。

**iPhone Dev Center サイト** Apple Computer の Web サイト (<http://developer.apple.com/iphone/>) では、次のことができます。

- iPhone デベロッパーの登録申請。
- iPhone 開発用証明書、プロビジョニングプロファイルおよびアプリケーション ID (以下に定義) の管理および作成。
- App Store へのアプリケーションの送信。

**iPhone 開発用証明書** アプリケーション開発を目的とする開発者であることを証明するために使用します。

このファイルは Apple から入手します。この証明書は、ActionScript 3.0 を使用して作成する iPhone アプリケーションに署名するために、P12 証明書ファイルに変換します。「[P12 証明書ファイル](#)」を参照してください。

開発コンピューターで Flash Professional CS5 のアプリケーションのデバッグやテストを行うだけの場合は、iPhone 開発用証明書は不要です。ただし、iPhone でアプリケーションをインストールおよびテストする場合は、証明書が必要です。

開発用証明書は、配布証明書とは異なります。配布証明書はアプリケーションの最終版のビルドに使用します。開発したアプリケーションの最終版が完成した際には、配布証明書を Apple から入手します。

**証明書署名要求** 開発用証明書を生成するために使用される個人情報を含むファイル。CSR ファイルとも呼ばれています。

**プロビジョニングプロファイル** iPhone アプリケーションをテストまたは配布できるようにするファイル。プロビジョニングプロファイルは Apple から入手します。プロビジョニングプロファイルは、特定の開発用証明書、アプリケーション ID、1 つ以上のデバイス ID に割り当てられます。次のように、異なるタイプのプロビジョニングプロファイルがあります。

- **開発プロビジョニングプロファイル** — アプリケーションのテスト版を開発者の iPhone にインストールするために使用されます。
- **テストプロビジョニングプロファイル** — アドホックプロビジョニングプロファイルとも呼ばれます。アプリケーションのテスト版を複数のユーザー（および iPhone ユニット）に配布するために使用されます。ユーザーは、このプロビジョニングプロファイルとテストアプリケーションを使用して、App Store に送信せずにアプリケーションをテストできます。注：複数のデバイスにテストアプリケーションを配布する際にも、開発プロビジョニングプロファイルを使用します。
- **配布プロビジョニングプロファイル** — iPhone アプリケーションを構築してアプリケーションを App Store に送信するために使用されます。

**アプリケーション ID** 特定の開発者からの 1 つの iPhone アプリケーション（または複数のアプリケーション）を識別する一意の文字列。アプリケーション ID は iPhone Dev Center サイトで作成します。各プロビジョニングプロファイルには、関連するアプリケーション ID またはアプリケーション ID パターンが含まれています。アプリケーションを開発するときは、このアプリケーション ID（またはパターン）を使用します。アプリケーション ID は、Flash Professional CS5 の iPhone 設定ダイアログボックス（またはアプリケーション記述ファイル）で使用します。

iPhone Dev Center のアプリケーション ID は、バンドルシード ID の後にバンドル識別子が続く組み合わせで構成されます。バンドルシード ID は、Apple がアプリケーション ID に割り当てる文字列（5RM86Z4DJM など）です。バンドル識別子は、選択したドメイン名の逆 DNS 形式の文字列です。バンドル識別子の末尾がアスタリスク（\*）の場合は、ワイルドカードアプリケーション ID であることを示しています。以下はその例です。

- 5RM86Z4DJM.com.example.helloWorld
- 96LPVWEASL.com.example.\*（ワイルドカードアプリケーション ID）

iPhone Dev Center には 2 種類のアプリケーション ID があります。

- **ワイルドカードアプリケーション ID - iPhone Dev Center** で、末尾がアスタリスク（\*）になっているアプリケーション ID（96LPVWEASL.com.myDomain.\* または 96LPVWEASL.\* など）。プロビジョニングプロファイルで、この種類のアプリケーション ID を使用している場合は、パターンに一致するアプリケーション ID を使用したテストアプリケーションを作成できます。アプリケーションのアプリケーション ID を指定するときは、アスタリスクを有効な文字列で置き換えます。例えば、iPhone Dev Center サイトで 96LPVWEASL.com.example.\* をアプリケーション ID に指定している場合は、com.example.foo や com.example.bar などアプリケーションのアプリケーション ID として使用できます。
- **固有のアプリケーション ID - アプリケーションで使用する固有のアプリケーション ID を定義します。** iPhone Dev Center では、これらのアプリケーション ID の末尾はアスタリスクではありません。例えば、96LPVWEASL.com.myDomain.myApp のようになっています。プロビジョニングプロファイルで、この種類のアプリケーション ID を使用している場合、アプリケーションとアプリケーション ID は正確に一致する必要があります。例えば、iPhone Dev Center サイトで 96LPVWEASL.com.example.helloWorld をアプリケーション ID に指定している場合は、com.example.foo をアプリケーションのアプリケーション ID として使用する必要があります。

アプリケーションを開発するときに、Flash Professional CS5 の iPhone 設定ダイアログボックスまたはアプリケーション記述ファイルでアプリケーション ID を指定します。アプリケーション ID について詳しくは、15 ページの「[Flash Professional CS5 における iPhone アプリケーションプロパティの設定](#)」の「「デプロイ」タブ」、または 17 ページの「[アプリケーション記述ファイルにおける iPhone アプリケーションプロパティの設定](#)」を参照してください。

**重要：**アプリケーション ID を指定する場合、アプリケーション ID のバンドルシード ID の部分は無視してください。例えば、Apple でアプリケーション ID が 96LPVWEASL.com.example.bob.myApp と表示されている場合は、96LPVWEASL を無視し、com.example.bob.myApp をアプリケーション ID として使用します。Apple でアプリケー

ション ID が 5RM86Z4DJM.\* と表示されている場合は、5RM86Z4DJM を無視します（これはワイルドカードアプリケーション ID です）。

プロビジョニングプロファイルに関連付けられているアプリケーション ID（またはアプリケーション ID のワイルドカードパターン）は、iPhone Dev Center (<http://developer.apple.com/iphone>) で確認できます。iPhone デベロッパープログラムポータルのプロビジョニングセクションに移動してください。

**P12 証明書ファイル** P12 ファイル（拡張子が .p12 のファイル）は、一種の証明書ファイル（個人情報交換ファイル）です。Packager for iPhone は、このタイプの証明書を使用して iPhone アプリケーションを構築します。Apple から入手する開発用証明書は、この形式の証明書に変換します。

**一意のデバイス ID** 特定の iPhone を識別する一意のコード。UDID またはデバイス ID とも呼ばれます。

## 開発ワークフローの概要

iPhone 用アプリケーションを開発するときは、次の手順に従います。

- 1 アドビから Flash Professional CS5 をインストールします。
- 2 iTunes をインストールします。
- 3 Apple から開発ファイルを手入します。これには、開発用証明書とプロビジョニングプロファイルが含まれます。4 ページの「[Apple から開発ファイルを手入](#)」を参照してください。
- 4 開発用証明書を P12 証明書ファイルに変換します。Flash CS5 では、証明書は P12 証明書である必要があります。4 ページの「[Apple から開発ファイルを手入](#)」を参照してください。
- 5 iTunes を使用して、プロビジョニングプロファイルを iPhone に関連付けます。
- 6 Flash Professional CS5 でアプリケーションを記述します。

iPhone アプリケーション用にコードをデザインおよび最適化するためのベストプラクティスについて理解することが重要です。35 ページの「[iPhone アプリケーションのデザイン上の考慮事項](#)」を参照してください。

また、一部の ActionScript 3.0 API は、iPhone では制限されているか、サポートされていません。28 ページの「[ActionScript 3.0 API によるモバイルデバイスのサポート](#)」を参照してください。

Flash Builder 4.0 を使用して、アプリケーション用の ActionScript 3.0 コードを編集することもできます。

開発コンピューターで、Flash Professional CS5 を使用してアプリケーションをテストできます。

- 7 アプリケーションのアイコンアートと起動画面アートを作成します。各 iPhone アプリケーションには、そのアプリケーションをユーザーが識別できるようにするためのアイコンのセットが含まれています。iPhone では、プログラムのロード時に起動画面イメージが表示されます。13 ページの「[iPhone のアイコンと起動画面イメージ](#)」を参照してください。
- 8 iPhone 設定を編集します。これらの設定には、次のものが含まれます。
  - アプリケーションの ID（ファイル名、アプリケーション名、バージョン番号およびアプリケーション ID を含む）
  - アプリケーションのソースアイコンアートの場所
  - アプリケーションに割り当てられている P12 証明書およびプロビジョニングプロファイル
  - アプリケーションの起動時の縦横比

Flash Professional CS5 では、iPhone 設定ダイアログボックスでこれらの設定を編集できます。詳しくは、15 ページの「[Flash Professional CS5 における iPhone アプリケーションプロパティの設定](#)」を参照してください。

アプリケーション記述ファイルで直接これらの設定を編集することもできます。詳しくは、17 ページの「[アプリケーション記述ファイルにおける iPhone アプリケーションプロパティの設定](#)」を参照してください。

- 9 Packager for iPhone を使用して IPA ファイルをコンパイルします。21 ページの「[iPhone アプリケーションインストーラー \(IPA\) ファイルのコンパイル](#)」を参照してください。

**10** iPhone でアプリケーションをインストールおよびテストします。iTunes を使用して IPA ファイルをインストールします。

個別の配布の際には、上記の一般的な手順を繰り返し実行します。ただし、開発プロビジョニングプロファイルではなくテストプロビジョニングプロファイルを使用してください。アプリケーションの最終配布時には、配布プロビジョニングプロファイルを使用してこの手順を実行します（プロビジョニングプロファイルのタイプについては、1 ページの「用語集」を参照してください）。

アプリケーションの配布バージョンを構築した場合は、27 ページの「[App Store への iPhone アプリケーションの送信](#)」を参照してください。

基本的な iPhone アプリケーションの構築の簡単なチュートリアルは、8 ページの「[Flash Professional CS5 での Hello World iPhone アプリケーションの作成](#)」を参照してください。

## アドビから開発ツールを入手

ActionScript 3.0 を使用して iPhone アプリケーションを開発するには、Flash Professional CS5 が必要です。

**重要：** Packager for iPhone のバージョンを、Flash Professional CS5 付属のプレビューバージョンから更新する必要があります。Flash Professional CS5 で、ヘルプ/アップデートを選択してください。

Flash Builder 4 を使用して ActionScript コードを編集することもできます。Flash Builder 4 は、<http://www.adobe.com/jp/products/flashbuilder/> から入手できます

## Apple から開発ファイル入手

iPhone 用アプリケーションの開発と同様に、まず Apple から iPhone 開発ファイル入手します。iPhone 開発用証明書と、モバイルプロビジョニングプロファイル入手する必要があります。また、その他のプロビジョニングプロファイルも入手する必要があります。これらのファイルについては詳しくは、1 ページの「用語集」を参照してください。

**注意：**これらのファイルの入手は、アプリケーション開発プロセスの重要な要素です。ファイルの入手を完了してからアプリケーションを開発するようにしてください。開発ファイルの入手は単純なプロセスではありません。この説明および Apple iPhone Dev Center サイトの説明を入念にお読みください。

## iPhone 開発ファイルの入手と操作

iPhone 開発用証明書とプロビジョニングプロファイルを Apple から入手する必要があります。また、証明書を P12 証明書に変換する必要があります。

### iTunes のインストール

アプリケーションを iPhone にインストールするには、iTunes が必要です。また、iTunes を使用して、iPhone のデバイス ID を決定します。iPhone 開発用証明書を適用するときには、デバイス ID が必要になります。

### iPhone 開発用証明書の申請およびプロビジョニングプロファイルの作成

まだ登録していない場合は、Apple iPhone Dev Center サイト (<http://developer.apple.com/iphone/>) で iPhone デベロッパーとして登録します。

**注意：** iPhone 向け AIR アプリケーションを開発するのに、iPhone SDK および XCode は必要ありません。ただし、iPhone デベロッパーとして登録する必要があります。また、開発用証明書およびプロビジョニングプロファイルを手入する必要があります。

- 1 iPhone デベロッパーのアカウント ID を使用して iPhone Dev Center にログインします。
- 2 iPhone Dev Center で、iPhone 開発用証明書を申請（および購入）します。  
iPhone デベロッパープログラムのライセンス認証コードが含まれた電子メールメッセージが、Apple から届きます。
- 3 iPhone Dev Center に戻ります。デベロッパープログラムのライセンス認証の手順に従います（ライセンス認証コードの入力を求められたら、これを入力します）。
- 4 ライセンス認証コードが受け入れられたら、iPhone Dev Center の iPhone デベロッパープログラムポータルセクションに移動します。
- 5 証明書署名要求ファイルを作成します。このファイルを使用して、iPhone 開発用証明書を取得します。方法については、6 ページの「[証明書署名要求の生成](#)」を参照してください。
- 6 次の手順で、お使いの iPhone のデバイス ID (UDID) を入力するよう求められます。UDID は次の手順で iTunes から取得できます。
  - a USB ケーブルを使用して iPhone を接続します。次に、iTunes で iPhone の「概要」タブを選択します。
  - b iPhone デベロッパーセンターのサイトからプロビジョニングプロファイルをダウンロードしたら、これを iTunes に追加します。
  - c 次に、表示されるシリアル番号をクリックします。UDID が表示されます。Command+C キー (Mac) または Ctrl+C キー (Windows) を押し、UDID をクリップボードにコピーします。
- 7 プロビジョニングプロファイルと iPhone 開発用証明書を作成し、インストールします。  
iPhone Dev Center の指示に従います。iPhone デベロッパープログラムポータルセクションで指示を確認します。開発用証明書を取得してプロビジョニングプロファイルを作成するには、Development Provisioning Assistant の使用をお勧めします。  
XCode に関連する手順は無視してください。Flash Professional CS5 を使用して iPhone アプリケーションを開発するときは、XCode を使用する必要はありません。
- 8 iTunes で、ファイル/ライブラリに追加を選択します。次に、プロビジョニングプロファイル（ファイル名の拡張子が mobileprovision のファイル）を選択します。この後、iPhone と iTunes を同期します。  
この操作により、iPhone 上でこのプロビジョニングプロファイルに関連付けされているアプリケーションをテストできます。  
iTunes に特定のプロビジョニングプロファイルが追加されたかどうかを検証するには、そのプロビジョニングプロファイルをライブラリに追加します。既存のプロビジョニングプロファイルを置き換えるかどうかを尋ねるメッセージが表示された場合には、「キャンセル」ボタンを押してください（そのプロファイルは既にインストールされています）。また、次のように操作して iPhone にインストールされているプロビジョニングプロファイルを確認することができます。
  - a iPhone で「設定」アプリケーションを開きます。
  - b 「一般」カテゴリを開きます。
  - c 「プロファイル」をタップします。「プロファイル」のページに、インストールされているプロビジョニングプロファイルが一覧表示されます。
- 9 iPhone 開発用証明書ファイル (.cer ファイル) をまだダウンロードしていない場合は、ダウンロードします。  
Development Provisioning Assistant にこのファイルをダウンロードするためのリンクが含まれる可能性があります。または、Apple iPhone Dev Center サイト (<http://developer.apple.com/iphone/>) の「Provisioning Portal」の「Certificates」セクションでこのファイルを取得することもできます。



10 次に、iPhone 開発用証明書を P12 ファイルに変換します。方法については、7 ページの「[P12 ファイルへの開発用証明書の変換](#)」を参照してください。

これで、シンプルな Hello World アプリケーションを作成する準備が整いました。8 ページの「[Flash Professional CS5 での Hello World iPhone アプリケーションの作成](#)」を参照してください。

## 証明書署名要求の生成

開発用証明書を取得するには、証明書署名要求ファイルを生成して、Apple iPhone Dev Center サイトに送信する必要があります。

### Mac OS での証明書署名要求の生成

Mac OS では、キーチェーンアクセスアプリケーションを使用して、コード署名要求を作成できます。キーチェーンアクセスアプリケーションは、アプリケーションディレクトリのユーティリティサブディレクトリにあります。キーチェーンアクセスメニューで、証明書アシスタント/認証局に証明書を要求を選択します。

- 1 キーチェーンアクセスを開きます。
- 2 キーチェーンアクセスメニューで、「環境設定」を選択します。
- 3 環境設定ダイアログボックスで、「証明書」をクリックします。次に、「オンライン証明書状況プロトコル (OCSP)」および「証明書失効リスト (CRL)」を「切」に設定します。ダイアログボックスを閉じます。
- 4 キーチェーンアクセスメニューで、証明書アシスタント/認証局に証明書を要求を選択します。
- 5 iPhone デベロッパーのアカウント ID の電子メールアドレスおよび名前を入力します。CA の電子メールアドレスは入力しないでください。「要求の処理」には「ディスクに保存」を選択して、「続ける」ボタンをクリックします。
- 6 ファイル (CertificateSigningRequest.certSigningRequest) を保存します。
- 7 [iPhone デベロッパーサイト](#)で、Apple に CSR ファイルをアップロードします (「iPhone 開発用証明書の申請およびプロビジョニングプロファイルの作成」を参照してください)。

### Windows での証明書署名要求の生成

Windows での開発者にとっては、Mac コンピューターで iPhone 開発用証明書を入手するほうが簡単です。しかし、Windows コンピューターで証明書を入手することもできます。最初に、OpenSSL を使用して証明書の署名リクエスト (CSR ファイル) を作成します。

- 1 Windows コンピューターに OpenSSL をインストールします (<http://www.openssl.org/related/binaries.html> に移動します)。

また、Visual C++ 2008 再頒布可能ファイルもインストールする必要があります。このファイルは Open SSL のダウンロードページにリストされています (コンピューターに Visual C++ をインストールする必要はありません)。

- 2 Windows コマンドセッションを開き、OpenSSL の bin ディレクトリ (c:\OpenSSL\bin など) に移動します。
- 3 コマンドラインで次のように入力して秘密鍵を作成します。

```
openssl genrsa -out mykey.key 2048
```

この秘密鍵ファイルを保存します。このファイルは後で使用します。

OpenSSL を使用するとき、エラーメッセージを無視しないようにしてください。OpenSSL でエラーメッセージが生成されても、引き続きファイルが出力される場合があります。ただし、それらのファイルは使用できません。エラーが表示されたら、シンタックスを確認して、コマンドを再実行してください。

- 4 コマンドラインで次のように入力して CSR ファイルを作成します。

```
openssl req -new -key mykey.key -out CertificateSigningRequest.certSigningRequest -subj  
"/emailAddress=yourAddress@example.com, CN=John Doe, C=US"
```

電子メールアドレス、CN（証明書名）、C（国）を、ユーザー独自の値に置き換えます。

- 5 **iPhone デベロッパーサイト**で、Apple に CSR ファイルをアップロードします（「iPhone 開発用証明書の申請およびプロビジョニングプロファイルの作成」を参照してください）。

## P12 ファイルへの開発用証明書の変換

Flash Professional CS5 を使用して iPhone アプリケーションを開発するには、P12 証明書ファイルを使用する必要があります。この証明書は、Apple から入手する Apple iPhone 開発用証明書ファイルに基づいて生成します。

### Mac OS での iPhone 開発用証明書の P12 ファイルへの変換

Apple iPhone 証明書を Apple からダウンロードしたら、P12 証明書形式に書き出します。Mac® OS でこれを行う手順は次のとおりです。

- 1 アプリケーション / ユーティリティフォルダー内にあるキーチェーンアクセスアプリケーションを開きます。
- 2 証明書をキーチェーンにまだ追加していない場合は、ファイル / 読み込みを選択します。次に、Apple から取得した証明書ファイル（.cer ファイル）に移動します。
- 3 キーチェーンアクセスの「分類」で「鍵」を選択します。
- 4 使用している iPhone 開発用証明書に関連付けられている秘密鍵を選択します。  
秘密鍵は、ペアになっている iPhone Developer: <First Name> <Last Name> 公開証明書によって識別されます。
- 5 ファイル / 書き出すを選択します。
- 6 .p12（Personal Information Exchange）ファイル形式で鍵を保存します。
- 7 この鍵を別のコンピューターに読み込むときに使用するパスワードを作成するよう指示があります。

### Windows における P12 ファイルへの Apple 開発用証明書の変換

Flash CS5 を使用して iPhone アプリケーションを開発するには、P12 証明書ファイルを使用する必要があります。この証明書は、Apple から入手する Apple iPhone 開発用証明書ファイルに基づいて生成します。

- 1 Apple から入手した開発用証明書ファイルを PEM 証明書ファイルに変換します。OpenSSL の bin ディレクトリから、次のコマンドラインステートメントを実行します。

```
openssl x509 -in developer_identity.cer -inform DER -out developer_identity.pem -outform PEM
```

- 2 Mac コンピューターで、キーチェーンから秘密鍵を使用している場合は、秘密鍵を PEM キーに変換します。

```
openssl pkcs12 -nocerts -in mykey.p12 -out mykey.pem
```

- 3 これで、キーと PEM 版の iPhone 開発用証明書に基づいて、有効な P12 ファイルを生成できます。

```
openssl pkcs12 -export -inkey mykey.key -in developer_identity.pem -out iphone_dev.p12
```

Mac OS キーチェーンからキーを使用している場合は、前の手順で生成した PEM 版を使用します。それ以外の場合は、前に（Windows で）作成した OpenSSL キーを使用します。

## 証明書、デバイス ID、アプリケーション ID およびプロビジョニングプロファイルの管理

証明書、デバイス ID、アプリケーション ID およびプロビジョニングプロファイルは Apple iPhone Dev Center サイト (<http://developer.apple.com/iphone>) で管理できます。このサイトの iPhone デベロッパープログラムポータルセクションに移動してください。

- 開発用証明書を管理するには、「Certificates」リンクをクリックします。証明書の作成、ダウンロードおよび取り消しを行うことができます。証明書を作成するには、最初に証明書署名要求を作成する必要があります。詳しくは、6 ページの「[証明書署名要求の生成](#)」を参照してください。
- 開発用証明書のインストール先のデバイス一覧を管理するには、「Devices」リンクをクリックします。
- アプリケーション ID を管理するには、「App IDs」リンクをクリックします。プロビジョニングプロファイルは、作成時にアプリケーション ID にバインドされます。
- プロビジョニングプロファイルを管理するには、「Provisioning」リンクをクリックします。Development Provisioning Assistant を使用して開発用プロビジョニングプロファイルを作成することもできます。
- アプリケーションを App Store に送信するか、アドホック用アプリケーションを作成する場合は、「Distribution」リンクをクリックします。このセクションには iTunes Connect サイトへのリンクがあり、App Store にアプリケーションを送信する際に使用します。

## Flash Professional CS5 での Hello World iPhone アプリケーションの作成

**重要：**アプリケーションを作成する前に、必要な開発アプリケーションおよびファイルをダウンロードします。4 ページの「[アドビから開発ツールを入手](#)」および 4 ページの「[Apple から開発ファイルを入手](#)」を参照してください。

### Flash Professional CS5 プロジェクトの作成

iPhone アプリケーションは、Flash Professional CS5 で直接生成できます。

- 1 Flash CS5 を開きます。
- 2 ファイル／新規を選択します。
- 3 iPhone OS を選択します。
- 4 「OK」ボタンをクリックします。

### アプリケーションへのコンテンツの追加

次に、「Hello world!」というテキストをアプリケーションに追加します。

- 1 「テキストツール」を選択してステージをクリックします。
- 2 テキストフィールドの「プロパティ設定」で、「(TLF テキスト)」ではなく「クラシックテキスト」を選択します  
これは基本的なアプリケーションなので、「クラシックテキスト」が適切です。TLF テキストを使用するためには、他の設定を適用する必要があります。38 ページの「[フォントとテキスト入力](#)」を参照してください。
- 3 新規のテキストフィールドに、「Hello World!」と入力します。
- 4 選択ツールでテキストフィールドを選択します。

5 プロパティインスペクターを開いて、次の設定を行います。

- 文字/フォントファミリー: `_sans`
- 文字/サイズ: 50
- 位置/X: 20
- 位置/Y: 20

6 ファイルを保存します。

7 制御/ムービープレビュー/AIR Debug Launcher (モバイル) を使用を選択します。

Flash Professional CS5 は SWF コンテンツをコンパイルし、アプリケーションを AIR Debug Launcher (ADL) に表示します。これで、アプリケーションを簡単にプレビューできます。

## アプリケーションのアイコンアートと起動画面アートの作成

すべての iPhone アプリケーションには、iTunes アプリケーションと iPhone 画面のユーザーインターフェイスに表示されるアイコンがあります。

1 プロジェクトディレクトリ内にディレクトリを作成して、`icons` という名前を付けます。

2 `icons` ディレクトリに PNG ファイルを 3 つ作成します。それぞれに `Icon29.png`、`Icon57.png`、`Icon512.png` という名前を付けます。

3 PNG ファイルを編集して、アプリケーションに適したアートを作成します。ファイルは 29 x 29 ピクセル、57 x 57 ピクセル、512 x 512 ピクセルである必要があります。このテストでは、アートとしてべた塗りの正方形を使用します

すべての iPhone アプリケーションで、アプリケーションが iPhone にロードされる間、起動画面イメージが表示されます。起動画面イメージは PNG ファイルで定義します。

1 メインの開発ディレクトリに、`Default.png` という名前の PNG ファイルを作成します (このファイルは `icons` サブディレクトリに入れなくてください。必ず、大文字の D を使用した `Default.png` という名前を付けてください)。

2 幅 320 ピクセル、高さ 480 ピクセルになるようにファイルを編集します。ここでは、コンテンツは単純な白の長方形にしておきます (これは後から変更します)。

**注意:** Apple App Store にアプリケーションを送信するときは、PNG 形式ではなく JPG 形式の 512 ピクセルファイルを使用します。PNG 形式のファイルは、開発版のアプリケーションをテストするときに使用します。

これらのグラフィックについて詳しくは、13 ページの「[iPhone のアイコンと起動画面イメージ](#)」を参照してください。

## アプリケーションの設定の編集

**重要:** iPhone 開発のために必要な開発アプリケーションとファイルをまだダウンロードしていない場合は、これをダウンロードします。4 ページの「[Apple から開発ファイル入手](#)」を参照してください。

Flash Professional CS5 の iPhone 設定ダイアログボックスで、iPhone アプリケーションの様々な基本プロパティを定義します。

1 ファイル/iPhone 設定を選択します。

2 「一般」タブで、次の設定を行います。

- 出力ファイル: `HelloWorld.ipa`  
これは、生成される iPhone インストーラーファイルのファイル名です。
- アプリケーション名: `Hello World`  
iPhone のアプリケーションアイコンの下に表示されるアプリケーションの名前です。

- バージョン：1.0  
アプリケーションのバージョンです。
- 起動時の縦横比：縦長モード
- フルスクリーンモードにする：選択
- 自動回転を有効にする：選択解除
- レンダリング：CPU

その他のオプションである「GPU」と「自動」では、レンダリングにハードウェアアクセラレーションを使用します。この機能を使用すると、ハードウェアアクセラレーションを活用するように設計された、グラフィックスを多用しているアプリケーション（ゲームなど）のパフォーマンスを向上させることができます。詳しくは、35 ページの「ハードウェアアクセラレーション」を参照してください。

- 含めるファイル：このリストには、起動画面アートファイル（Default.png）を追加します。

**注意：**この Hello World の例では、この説明にある設定を変更しないでください。バージョン設定などの一部の設定には、特定の制限事項があります。これらの制限事項については、15 ページの「iPhone アプリケーション設定」で説明します。

### 3 「デプロイ」タブで、次の設定を行います。

- 証明書：Apple から入手した開発用証明書に基づいて .p12 証明書を探して選択します。  
この証明書は、ファイルに署名をする際に使用します。Apple iPhone 証明書を .p12 形式に変換する必要があります。詳しくは、4 ページの「[アドビから開発ツールを入手](#)」を参照してください。
- パスワード：証明書のパスワードを入力します。
- プロビジョニングファイル：Apple から入手した開発プロビジョニングファイルを参照して選択します。4 ページの「[Apple から開発ファイル入手](#)」を参照してください。
- アプリケーション ID：このフィールドが選択可能な場合は、Apple に提供したアプリケーション ID に一致するアプリケーション ID（com.example.as3.HelloWorld など）を入力できます。

アプリケーションを一意に識別するアプリケーション ID です。

このフィールドが選択可能でない場合は、プロビジョニングプロファイルが特定のアプリケーション ID にバインドされています。アプリケーション ID がフィールドに表示されます。

アプリケーション ID の指定について詳しくは、15 ページの「[Flash Professional CS5 における iPhone アプリケーションプロパティの設定](#)」の「「デプロイ」タブ」を参照してください。

### 4 「アイコン」タブの「アイコン」リストで、29 x 29 のアイコンをクリックします。次に、前に作成した 29 x 29 ピクセルの PNG ファイルの場所を指定します（9 ページの「[アプリケーションのアイコンアートと起動画面アートの作成](#)」を参照してください）。次に、57 x 57 ピクセルのアイコンと 512 x 512 ピクセルのアイコンの PNG ファイルを指定します。

### 5 「OK」ボタンをクリックします。

### 6 ファイルを保存します。

アプリケーション設定について詳しくは、15 ページの「[iPhone アプリケーション設定](#)」を参照してください。

## IPA ファイルのコンパイル

ここで、IPA インストーラーファイルをコンパイルできます。

- 1 ファイル／パブリッシュを選択します。
- 2 iPhone の設定ダイアログボックスで、「OK」ボタンをクリックします。

Packager for iPhone により、iPhone アプリケーションインストーラーファイル、HelloWorld.ipa ファイルがプロジェクトディレクトリに生成されます。IPA ファイルのコンパイルには数分かかる場合があります。

## iPhone へのアプリケーションのインストール

iPhone アプリケーションをテスト用に iPhone にインストールするには：

- 1 iTunes アプリケーションを開きます。
- 2 このアプリケーションのプロビジョニングプロファイルを iTunes にまだ追加していない場合は、追加します。iTunes で、ファイル／ファイルをライブラリに追加を選択します。次に、ファイルタイプ **mobileprovision** のプロビジョニングプロファイルを選択します。

開発 iPhone でアプリケーションをテストするには、開発プロビジョニングプロファイルを使用します。

後で、アプリケーションを iTunes Store に配布するときに、配布プロファイルを使用します。アプリケーションをアドホックで (iTunes Store を経由せずに複数のデバイスに) 配布するには、アドホックプロビジョニングプロファイルを使用します。

プロビジョニングプロファイルについては、4 ページの「[Apple から開発ファイル入手](#)」を参照してください。

- 3 一部のバージョンの iTunes では、同じバージョンのアプリケーションが既にインストールされている場合、アプリケーションは置き換えられません。この場合は、デバイスおよび iTunes のアプリケーションリストから、該当するアプリケーションを削除します。
- 4 アプリケーションの IPA ファイルをダブルクリックします。このファイルがアプリケーションのリストに表示されます。
- 5 iPhone を、コンピューターの USB ポートに接続します。
- 6 iTunes で、このデバイスの「アプリケーション」タブを確認し、インストールするアプリケーションのリストで該当するアプリケーションが選択されていることを確認します。
- 7 iTunes アプリケーションの左側のリストでデバイスを選択します。「同期」ボタンをクリックします。同期が完了すると、Hello World アプリケーションが iPhone に表示されます。

新しいバージョンがインストールされない場合は、iPhone および iTunes のアプリケーションリストから削除してから、この手順を再度実行します。現在インストールされているアプリケーションで、インストールしようとしているアプリケーションと同じアプリケーション ID とバージョンが使用されている場合に、この問題が発生することがあります。

アプリケーションのインストール時に iTunes でエラーが表示される場合は、23 ページの「[iPhone アプリケーションのインストール](#)」の「アプリケーションのインストールに関する問題のトラブルシューティング」を参照してください。

## 起動画面イメージの編集

アプリケーションをコンパイルする前に、Default.png ファイルを作成しました (9 ページの「[アプリケーションのアイコンアートと起動画面アートの作成](#)」を参照)。この PNG ファイルは、アプリケーションのロード中に、起動画面イメージの役割を果たします。iPhone でアプリケーションをテストしたときに、起動時にこの空の画面が表示されることに気づいた方もいるでしょう。

このイメージを、アプリケーション (「Hello World!」) のスタートアップスクリーンと一致するように変更する必要があります。

- 1 iPhone デバイスでアプリケーションを開きます。最初の「Hello World」というテキストが表示されたら、ホームボタン (画面の下) を長押しします。ホームボタンを押したまま、電源 / スリープボタン (iPhone の上部) を押します。これでスクリーンショットが撮影され、カメラロールに送信されます。
- 2 iPhoto またはその他の写真転送アプリケーションから、このスクリーンショットイメージを開発コンピューターに転送します (Mac OS では、イメージキャプチャアプリケーションも使用できます)。

スクリーンショットイメージは、次の手順で開発コンピューターにメールで送信することもできます。

- 写真アプリケーションを開きます。
- カメラロールを開きます。
- キャプチャしたスクリーンショットイメージを開きます。
- そのイメージをタップして、左下隅の転送（矢印） ボタンをタップします。「写真をメール」 ボタンをクリックしてイメージを自分宛に送信します。

**3** Default.png ファイル（開発ディレクトリ内）を、PNG 形式のスクリーンショットイメージに置き換えます。

**4** アプリケーションを再コンパイルして（10 ページの「[IPA ファイルのコンパイル](#)」を参照）、iPhone に再インストールします。

これで、アプリケーションのロード時に新しいスタートアップスクリーンが使用されます。

**注意：**サイズが正しければ（320 x 480 ピクセル）、どのようなアートでも、Default.png ファイルとして作成できます。ただし、多くの場合には、Default.png イメージには、アプリケーションの初期状態と同じイメージを使用することをお勧めします。

## 第2章：iPhone アプリケーションのコンパイルとデバッグ

Packager for iPhone を使用して iPhone アプリケーションをコンパイルできます。Flash Professional CS5 には Packager for iPhone が付属しています。

アプリケーションは開発コンピューターでデバッグできます。また、デバッグ版を iPhone にインストールし、Flash Professional CS5 で trace() 出力を受け取ることができます。

iPhone アプリケーションの構築方法を最初から最後まで説明するチュートリアルについては、8 ページの「[Flash Professional CS5 での Hello World iPhone アプリケーションの作成](#)」を参照してください。

### iPhone のアイコンと起動画面イメージ

すべての iPhone アプリケーションには、iTunes アプリケーションと iPhone のユーザーインターフェイスに表示されるアイコンがあります。

#### iPhone アプリケーションアイコン

iPhone アプリケーション用に次のアイコンを定義します。

- 29 x 29 ピクセルのアイコン - このアイコンは iPhone および iPod Touch のスポットライト検索結果で使用されます。
- 48 x 48 ピクセルのアイコン - このアイコンは iPad のスポットライト検索結果で使用されます。
- 57 x 57 ピクセルのアイコン - このアイコンは iPhone および iPod Touch のホーム画面に表示されます。
- 72 x 72 ピクセルのアイコン (オプション) - このアイコンは iPad のホーム画面に表示されます。
- 512 x 512 ピクセルのアイコン - このアイコンは iTunes に表示されます。512 ピクセルの PNG ファイルが使用されるのは、開発版のアプリケーションをテストする場合のみです。Apple App Store に最終版のアプリケーションを送信する場合、512 イメージは別途 JPG ファイルとして送信します。このイメージは IPA には含まれません。

Flash Professional CS5 では、iPhone 設定ダイアログボックスの「アイコン」タブでこれらのアイコンを追加します。15 ページの「[Flash Professional CS5 における iPhone アプリケーションプロパティの設定](#)」を参照してください。

アプリケーション記述ファイルにアイコンの場所を追加することもできます。

```
<icon>
  <image29x29>icons/icon29.png</image29x29>
  <image57x57>icons/icon57.png</image57x57>
  <image72x72>icons/icon72.png</image72x72>
  <image512x512>icons/icon512.png</image512x512>
</icon>
```

アイコンのグレア効果は、iPhone によって追加されます。効果をソースイメージに追加する必要はありません。デフォルトのグレア効果を取り消すには、アプリケーション記述ファイルの InfoAdditions エレメントに以下を追加します。

```
<InfoAdditions>
  <![CDATA[
    <key>UIPrerenderedIcon</key>
    <true/>
  ]]>
</InfoAdditions>
```

17 ページの「[アプリケーション記述ファイルにおける iPhone アプリケーションプロパティの設定](#)」を参照してください。



### 起動画面アート (Default.png)

すべての iPhone アプリケーションで、アプリケーションが iPhone にロードされる間、起動画面イメージが表示されます。起動画面イメージは Default.png という名前の PNG ファイルで定義します。メインの開発ディレクトリに、Default.png という名前の PNG ファイルを作成します（このファイルはサブディレクトリに入れないでください。必ず、大文字の D を使用した Default.png という名前を付けてください）。

Default.png ファイルは、アプリケーションの起動時のモード（縦長または横長）やフルスクリーンモードであるかどうかにかかわらず、320 x 480 ピクセル（幅 x 高さ）です。

アプリケーションの起動時のモードが横長モードの場合は、縦長モードのアプリケーションと同じサイズを使用します。つまり、320 x 480 ピクセル（幅 x 高さ）です。ただし、PNG ファイルでア트워크を反時計回りに 90° 回転させます。横長モードでは、PNG アートの左側が iPhone 画面の上部に該当します（アプリケーションの起動時のモード設定については、15 ページの「[iPhone アプリケーション設定](#)」を参照してください）。

フルスクリーンモード以外のアプリケーションでは、デフォルトのイメージアートの上部 20 ピクセルは無視されます。iPhone では、デフォルトイメージの上部、幅 20 ピクセルの矩形部分に、ステータスバーが表示されます。横長モードのアプリケーションでは、この領域は、Default.png ファイルの左側、幅 20 ピクセルの矩形部分（横長モードで上部に表示されている）に該当します。縦長モードのアプリケーションでは、この領域は、Default.png ファイルの上部、幅 20 ピクセルの矩形部分に該当します。

ほとんどのアプリケーションでは、Default.png イメージはアプリケーションの起動画面と同じになります。アプリケーションの起動画面のスクリーンショットを作成するには：

- 1 アプリケーションを iPhone で開きます。ユーザーインターフェイスの最初の画面が表示されたら、画面の下にあるホームボタンを長押しします。ホームボタンを押したまま、電源 / スリープボタン (iPhone デバイスの上部) を押しします。これでスクリーンショットが撮影され、カメラロールに送信されます。
- 2 iPhoto またはその他の写真転送アプリケーションから、このスクリーンショットイメージを開発コンピューターに転送します (Mac OS では、イメージキャプチャアプリケーションも使用できます)。

スクリーンショットイメージは、次の手順で開発コンピューターにメールで送信することもできます。

- 写真アプリケーションを開きます。
- カメラロールを開きます。
- キャプチャしたスクリーンショットイメージを開きます。
- そのイメージをタップして、左下隅の転送 (矢印) ボタンをタップします。「写真をメール」ボタンをクリックしてイメージを自分宛に送信します。

**注意：**サイズさえ正しければ、どのようなアートでも、Default.png ファイルとして作成できます。ただし、多くの場合には、Default.png イメージには、アプリケーションの初期状態と同じイメージを使用することをお勧めします。

アプリケーションが複数の言語にローカライズされる場合は、Default.png イメージにテキストを含めないでください。Default.png は静的であり、テキストが他の言語に適合しません。

Flash Professional CS5 では、iPhone 設定ダイアログボックスの「含めるファイル」リストに、Default.png ファイルを必ず追加してください。15 ページの「[Flash Professional CS5 における iPhone アプリケーションプロパティの設定](#)」を参照してください。

コマンドラインで PFI アプリケーションを使用してコンパイルする場合は、含まれているアセットのリスト内のこのファイルを参照してください。21 ページの「[コマンドラインからの iPhone アプリケーションインストーラーファイルの作成](#)」を参照してください。

## iPhone アプリケーション設定

アプリケーション設定には次の設定が含まれます。

- アプリケーション名
- IPA ファイル名
- アプリケーションのバージョン
- アプリケーションの起動画面の方向、および iPhone 回転時に画面の方向が自動的に回転するかどうか
- 起動画面がフルスクリーンかどうか
- アプリケーションのアイコンに関する情報
- ハードウェアアクセラレーションに関する情報

Flash Professional CS5 でアプリケーション設定を編集できます。

アプリケーション記述ファイルでアプリケーション設定を編集することもできます。アプリケーション記述ファイルは、アプリケーションの設定を含む XML ファイルです。

## Flash Professional CS5 における iPhone アプリケーションプロパティの設定

Flash Professional CS5 の iPhone 設定ダイアログボックスを使用すると、iPhone アプリケーションの様々な基本プロパティを定義できます。

iPhone 設定ダイアログボックスを開くには：

- ❖ ファイル / iPhone 設定を選択します。

### 「一般」タブ

「一般」タブには、iPhone に関連する次の設定が含まれています。

- 出力ファイル - iPhone のアプリケーションアイコンの下に表示されるアプリケーションの名前です。出力ファイル名にはプラス記号 (+) を含めないでください。
- アプリケーション名 - iPhone のアプリケーションアイコンの下に表示されるアプリケーションの名前です。アプリケーション名にはプラス記号 (+) を含めないでください。
- バージョン - インストールしているアプリケーションのバージョンを確認できます。このバージョン番号は、iPhone アプリケーションの CFBundleVersion として使用されます。この番号は、nnnnn[.nn[.nn]] のような形式になります。ここで n は 0 ~ 9 の数値、角括弧はオプションの構成要素を示します。例えば、1、1.0、1.0.1 のようになります。iPhone バージョンには数値および小数点のみを含める必要があります。iPhone バージョンに含めることのできる小数点は 2 つまでです。
- 起動時の縦横比 - アプリケーションの起動時の縦横比（縦長モードまたは横長モード）
- フルスクリーンモードにする - アプリケーションでフルスクリーンを使用するか、iPhone ステータスバーを表示するか。
- 自動回転を有効にする - iPhone の向きを変更したときに、アプリケーションの表示コンテンツを自動的に回転させるには、このオプションを選択します。

自動回転の使用時に最適な結果を得るために、ActionScript コードを追加して Stage の align プロパティを次のように設定します。

```
stage.align = StageAlign.TOP_LEFT;  
stage.scaleMode = StageScaleMode.NO_SCALE;
```

- レンダリング - iPhone での表示オブジェクトのレンダリング方法。
  - CPU - アプリケーションは CPU を使用してすべての表示オブジェクトをレンダリングします。ハードウェアアクセラレーションは使用されません。
  - GPU - アプリケーションは iPhone の GPU を使用してビットマップを合成します。
  - 自動 - この機能は実装されていません。詳しくは、35 ページの「ハードウェアアクセラレーション」を参照してください。
- 含めるファイル - iPhone アプリケーションにパッケージ化するすべてのファイルとディレクトリを追加します。メイン SWF ファイルおよびアプリケーション記述ファイルはデフォルトで含まれています。他に必要なアセットがある場合は、それらを「含めるファイル」リストに追加します。起動画面アートファイル (Default.png) は、「含めるファイル」リストに必ず追加してください。

### 「デブロイ」タブ

「デブロイ」タブには、アプリケーションの署名設定およびコンパイル設定が含まれます。

- iPhone 電子署名 - P12 証明書ファイルと証明書のパスワードを指定します。Apple iPhone 証明書を .p12 形式に変換する必要があります。詳しくは、4 ページの「Apple から開発ファイル入手」を参照してください。
- プロビジョニングファイル - Apple から入手した、このアプリケーション用のプロビジョニングファイルを指定します。詳しくは、4 ページの「Apple から開発ファイル入手」を参照してください。
- アプリケーション ID - アプリケーションを一意に識別します。プロビジョニングファイルが特定のアプリケーション ID に関連付けられている場合は、Flash Professional CS5 によってこのフィールドが設定されます。ユーザーは編集できません。それ以外の場合は、プロビジョニングプロファイルで複数の (ワイルドカード) アプリケーション ID を使用できます。Apple に提供したアプリケーション ID のワイルドカードパターンに一致するアプリケーション ID を指定します。
  - 例えば、Apple アプリケーション ID が com.myDomain.\* の場合、iPhone 設定ダイアログボックスのアプリケーション ID は、com.myDomain で始まる必要があります (例えば、com.myDomain.myApp または com.myDomain.app22 などです)。
  - Apple アプリケーション ID が \* の場合は、有効な文字であればどのような文字列でも iPhone 設定ダイアログボックスのアプリケーション ID に使用できます。

プロビジョニングプロファイルに関連付けられている Apple アプリケーション ID (またはアプリケーション ID のワイルドカードパターン) は、iPhone Dev Center (<http://developer.apple.com/iphone>) で確認できます。iPhone デベロッパープログラムポータルのプロビジョニングセクションに移動してください。

**重要:** Apple アプリケーション ID の先頭の文字列は無視してください。Apple ではこの文字列をバンドルシード ID と呼んでいます。例えば、Apple でアプリケーション ID が 96LPVWEASL.com.example.bob.myApp と表示されている場合は、96LPVWEASL を無視し、com.example.bob.myApp をアプリケーション ID として使用します。Apple でアプリケーション ID が 5RM86Z4DJM.\* と表示されている場合は、5RM86Z4DJM を無視します (これはワイルドカードアプリケーション ID です)。

- iPhone デプロイタイプ:
  - クイックパブリッシュ - デバイスでのテスト用 - 開発 iPhone でアプリケーションのテスト用のバージョンを迅速にコンパイルするには、このオプションを選択します。
  - デバイスのデバッグを目的としたクイックパブリッシュ - 開発 iPhone でアプリケーションのデバッグ用のバージョンを迅速にコンパイルするには、このオプションを選択します。このオプションを使用すると、Flash Professional CS5 のデバッガーは、iPhone アプリケーションから trace() を受け取ることができます (24 ページの「iPhone アプリケーションのデバッグ」を参照してください)。
  - デプロイ - アドホック - アドホックデプロイ用にアプリケーションを作成するには、このオプションを選択します。Apple iPhone デベロッパーセンターを参照してください。

- デプロイ - Apple App Store - Apple App Store にデプロイする最終版の IPA ファイルを作成するには、このオプションを選択します。

#### 「アイコン」タブ

「アイコン」タブでは、29 x 29 ピクセルのアイコンイメージ、48 x 48 ピクセルのアイコンイメージ、57 x 57 ピクセルのアイコンイメージ、72 x 72 ピクセルのアイコンイメージ、および 512 x 512 ピクセルのアイコンイメージの場所を指定します。13 ページの「[iPhone のアイコンと起動画面イメージ](#)」を参照してください。

**注意：**48 x 48 ピクセルおよび 72 x 72 ピクセル用のオプションは、Flash Professional CS5 に付属する Packager for iPhone Preview バージョンには含まれていません。Flash Professional CS5 でヘルプ/アップデートを選択してこれらのオプションを追加してください。

## アプリケーション記述ファイルにおける iPhone アプリケーションプロパティの設定

アプリケーション記述ファイルは、アプリケーション名、バージョン、著作権、その他の設定など、アプリケーション全体のプロパティを含む XML ファイルです。

Flash Professional CS5 では、iPhone 設定ダイアログボックスの設定に基づいて、アプリケーション記述ファイルが生成されます。ただし、テキストエディターでアプリケーション記述ファイルを編集することもできます。生成されたアプリケーション記述ファイルには、プロジェクト名に「-app.xml」を追加した名前が付けられます。例えば、HelloWorld プロジェクトには、HelloWorld-app.xml という名前のアプリケーション記述ファイルが生成されます。Flash Professional CS5 の iPhone 設定ダイアログボックスでサポートされていない設定を定義する場合は、アプリケーション記述ファイルを編集します。例えば、InfoAdditions エlement を定義して、アプリケーションの info.Plist 設定を定義することができます。

**重要：**Flash Professional CS5 のダイアログボックスが開いている間は、アプリケーション記述ファイルを編集しないでください。アプリケーション記述ファイルの変更内容は、iPhone 設定ダイアログボックスを開く前に保存します。

アプリケーション記述ファイルの例を次に示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<application xmlns="http://ns.adobe.com/air/application/2.0">
  <id>com.example.HelloWorld</id>
  <filename>HelloWorld</filename>
  <name>Hello World</name>
  <version>v1</version>
  <initialWindow>
    <renderMode>gpu</renderMode>
    <content>HelloWorld.swf</content>
    <fullScreen>true</fullScreen>
    <aspectRatio>portrait</aspectRatio>
    <autoOrients>true</autoOrients>
  </initialWindow>
  <supportedProfiles>mobileDevice desktop</supportedProfiles>
  <icon>
    <image29x29>icons/icon29.png</image29x29>
    <image57x57>icons/icon57.png</image57x57>
    <image512x512>icons/icon512.png</image512x512>
  </icon>
  <iPhone>
    <InfoAdditions>
      <![CDATA[
        <key>UIStatusBarStyle</key>
        <string>UIStatusBarStyleBlackOpaque</string>
        <key>UIRequiresPersistentWiFi</key>
        <string>NO</string>
      ]]>
    </InfoAdditions>
  </iPhone>
</application>
```

このアプリケーション記述ファイルの設定の詳細は次のとおりです。

- <application> エレメントには、iPhone アプリケーションの構築に必要な AIR 2.0 の名前空間を指定する必要があります。

```
<application xmlns="http://ns.adobe.com/air/application/2.0">
```

- <id> エレメント：

**<id>com.example.as3.HelloWorld</id>** アプリケーションを一意に識別するアプリケーション ID。推奨される形式は、「com.company.AppName」などのピリオド区切りの逆 DNS スタイルのストリングです。コンパイラーは、この値を、iPhone アプリケーションのバンドル ID として使用します。

プロビジョニングファイルが特定のアプリケーション ID に関連付けられている場合は、該当するアプリケーション ID をこのエレメントで使用します。Apple アプリケーション ID の先頭部分（バンドルシード ID と呼ばれる）に割り当てられている文字列は無視します。例えば、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 96LPVWEASL.com.example.bob.myApp の場合、アプリケーション記述ファイルでは、com.example.bob.myApp をアプリケーション ID として使用します。

プロビジョニングプロファイルで複数の（ワイルドカード）アプリケーション ID を使用できる場合、そのアプリケーション ID の末尾はアスタリスクです（5RM86Z4DJM.\* など）。Apple に提供したアプリケーション ID のワイルドカードパターンに一致するアプリケーション ID を指定します。

- 例えば、Apple アプリケーション ID が com.myDomain.\* の場合、アプリケーション記述ファイルのアプリケーション ID は、com.myDomain で始まる必要があります。com.myDomain.myApp または com.myDomain.app22 のようなアプリケーション ID を指定できます。
- Apple アプリケーション ID が \* の場合は、有効な文字であればどのような文字列でもアプリケーション記述ファイルのアプリケーション ID に使用できます。

プロビジョニングプロファイルに関連付けられている Apple アプリケーション ID (またはアプリケーション ID のワイルドカードパターン) は、iPhone Dev Center (<http://developer.apple.com/iphone>) で確認できます。iPhone デベロッパープログラムポータルのプロビジョニングセクションに移動してください。

**重要:** Apple アプリケーション ID の先頭の文字列は無視してください。Apple ではこの文字列をバンドルシード ID と呼んでいます。例えば、Apple でアプリケーション ID が 5RM86Z4DJM.\* と表示されている場合は、5RM86Z4DJM を無視します (これはワイルドカードアプリケーション ID です)。ご使用のアプリケーション ID が Apple では 96LPVWEASL.com.example.bob.myApp と表示されている場合は、96LPVWEASL を無視し、com.example.bob.myApp をアプリケーション ID として使用します。

- <filename> エlement:

**<filename>HelloWorld</filename>** iPhone インストーラーファイルに使用する名前。ファイル名にはプラス記号 (+) を含めないでください。

- <name> エlement:

**<name>Hello World</name>** iTunes アプリケーションおよび iPhone に表示されるアプリケーションの名前。名前にはプラス記号 (+) を含めないでください。

- <version> エlement:

**<version>1.0</version>** インストールしているアプリケーションのバージョンをユーザーが確認できるようにします。このバージョン番号は、iPhone アプリケーションの CFBundleVersion として使用されます。この番号は、nnnnn[.nn[.nn]] のような形式になります。ここで n は 0 ~ 9 の数値、角括弧はオプションの構成要素を示します。例えば、1、1.0、1.0.1 のようになります。iPhone バージョンには数値および小数点のみを含める必要があります。iPhone バージョンに含めることのできる小数点は 2 つまでです。

- <initialWindow> エlementには、アプリケーションの初期画面のプロパティを指定する、次の子エlementが含まれます。

**<content>HelloWorld.swf</content>** iPhone アプリケーションにコンパイルされるルート SWF ファイルを識別します。

**<visible>>true</visible>** これは必須の設定です。

**<fullScreen>>true</fullScreen>** アプリケーションが iPhone の画面全体を使用することを指定します。

**<aspectRatio>portrait</aspectRatio>** アプリケーションの起動時の縦横比が横長モードではなく縦長モードであることを指定します。アプリケーションの初期ウィンドウを定義する Default.png ファイルは、この設定にかかわらず、幅 320 ピクセル、高さ 480 ピクセルにする必要があります (13 ページの「[iPhone のアイコンと起動画面イメージ](#)」を参照してください)。

**<autoOrients>>true</autoOrients>** (オプション) デバイス自体の向きが変わったときに、アプリケーションのコンテンツの向きを自動的に変えるかどうかを指定します。デフォルト値は true です。表示方向の自動変更は、Stage オブジェクトによって送出される orientationChanging イベントの preventDefault() メソッドを呼び出すことによってキャンセルできます。詳しくは、「[スクリーンの向きの設定および検出](#)」を参照してください。

自動回転の使用時に最適な結果を得るために、Stage の align プロパティを次のように設定します。

```
stage.align = StageAlign.TOP_LEFT;
stage.scaleMode = StageScaleMode.NO_SCALE;
```

**<renderMode>gpu</renderMode>** (オプション) アプリケーションによって使用されるレンダリングモードです。次の 3 つの設定が可能です。

- cpu - アプリケーションは CPU を使用してすべての表示オブジェクトをレンダリングします。ハードウェアアクセラレーションは使用されません。
- gpu - アプリケーションは iPhone の GPU を使用してビットマップを合成します。
- auto - この機能は実装されていません。

詳しくは、35 ページの「ハードウェアアクセラレーション」を参照してください。

- <profiles> エレメント :

<profiles>mobileDevice</profiles> モバイルデバイスプロファイルにコンパイルされるようにアプリケーションを制限します。このプロファイルで現在サポートされるのは、iPhone アプリケーションのみです。次の 3 つのプロファイルがサポートされます。

- desktop - デスクトップ AIR アプリケーションです。
- extendedDesktop - ネイティブプロセス API のサポートを含むデスクトップ AIR アプリケーションです。
- mobileDevice - モバイルデバイス用の AIR アプリケーションです。現在サポートされているモバイルデバイスは iPhone のみです。

アプリケーションを特定のプロファイルに限定すると、他のプロファイルにコンパイルされるのを防ぐことができます。プロファイルを指定しない場合は、これらの任意のプロファイル用にアプリケーションをコンパイルできます。

<profiles> エレメントで、複数のプロファイルをスペースで区切って指定することができます。

必ず、mobileDevice をサポートされるプロファイルとして含めてください（または、<profiles> エレメントを空のままにします）。

- <icon> エレメントには、アプリケーションで使用するアイコンを指定する、次の子エレメントが含まれます。

<image29x29>icons/icon29.png</image29x29> スポットライト検索結果で使用するイメージです。

<image48x48>icons/icon48.png</image48x48> iPad でのスポットライト検索結果で使用するイメージです。

<image57x57>icons/icon57.png</image57x57> iPhone および iPod Touch のホーム画面で使用するイメージです。

<image72x72>icons/icon72.png</image72x72> iPad ホーム画面で使用するイメージです。

<image512x512>icons/icon512.png</image512x512> iTunes アプリケーションで使用するイメージです。

Packager for iPhone ツールは、アプリケーション記述ファイル内で示される 29、57 および 512 の各種アイコンを使用します。各アイコンはそれぞれ、Icon-Small.png、Icon.png、および iTunesArtwork と呼ばれるファイルにコピーされます。このコピーが行われないようにするには、これらのファイルを直接パッケージします。アプリケーション記述ファイルを格納しているディレクトリにファイルを配置し、正しいファイル名とパスをリストして直接パッケージします。

512 イメージは、内部テスト専用です。アプリケーションを Apple に送信するときは、512 イメージは別途送信します。このイメージは IPA には含まれません。512 イメージが iTunes で正常に表示されるように指定してから送信してください。

- <iPhone> エレメントには、iPhone に固有の設定を指定するための、次の子エレメントが含まれます。

<InfoAdditions></InfoAdditions> アプリケーションの Info.plist 設定として使用するキーと値のペアを指定する、次の子エレメントが含まれます。

```
<![CDATA [  
  <key>UIStatusBarStyle</key>  
  <string>UIStatusBarStyleBlackOpaque</string>  
  <key>UIRequiresPersistentWiFi</key>  
  <string>NO</string>  
]]>
```

この例では、これらの子エレメントが、アプリケーションのステータスバーのスタイルを設定しています。また、アプリケーションに永続的な Wi-Fi アクセスが不要であることを示しています。

InfoAdditions の設定は、CDATA タグで囲まれています。

iPad をサポートするためには、UIDeviceFamily にキーおよび値の設定を含めてください。UIDeviceFamily 設定は string の配列です。各 string ではサポートするデバイスを定義します。<string>1</string> 設定は、iPhone および iPod Touch のサポートを定義します。<string>2</string> 設定は、iPad のサポートを定義します。<string>3</string> 設定は、tvOS のサポートを定義します。これらの設定の一方のみを指定した場合、そのデバイス群のみがサポートされます。例えば、次の設定では iPad のサポートのみに限定されます。

```
<key>UIDeviceFamily</key>
  <array>
    <string>2</string>
  </array>>
```

次の場合は、両方のデバイス群 (iPhone/iPod Touch と iPad) のサポートが設定されます。

```
<key>UIDeviceFamily</key>
<array>
  <string>1</string>
  <string>2</string>
</array>
```

Info.plist のその他の設定について詳しくは、Apple の開発ドキュメントを参照してください。

## iPhone アプリケーションインストーラー (IPA) ファイルのコンパイル

Packager for iPhone を使用して ActionScript 3.0 アプリケーションをコンパイルし、IPA インストーラーファイルを作成します。

### Flash Professional CS5 に付属する Packager for iPhone を使用した iPhone アプリケーションインストーラーファイルの作成

Flash Professional CS5 に付属する Packager for iPhone を使用するには：

- 1 ファイル/パブリッシュを選択します。
- 2 iPhone 設定ダイアログボックスで、すべての設定の値を指定したことを確認します。「デプロイ」タブで、正しいオプションを選択したことを確認します。15 ページの「[Flash Professional CS5 における iPhone アプリケーションプロパティの設定](#)」を参照してください。
- 3 「パブリッシュ」ボタンをクリックします。

Packager for iPhone によって、iPhone アプリケーションインストーラー (IPA) ファイルが生成されます。IPA ファイルのコンパイルには数分かかる場合があります。

Packager for iPhone をコマンドラインから実行することもできます。21 ページの「[コマンドラインからの iPhone アプリケーションインストーラーファイルの作成](#)」を参照してください。

### コマンドラインからの iPhone アプリケーションインストーラーファイルの作成

Packager for iPhone はコマンドラインから実行できます。Packager for iPhone は、SWF ファイルのバイトコードおよび他のソースファイルをネイティブ iPhone アプリケーションに変換します。

- 1 コマンドシェルまたは端末を開き、iPhone アプリケーションのプロジェクトフォルダーに移動します。
- 2 次に、pfi ツールを使用して IPA ファイルを作成します。次の構文を使用します。

```
pfi -package -target [ipa-test ipa-debug ipa-app-store ipa-ad-hoc] -provisioning-profile PROFILE_PATH SIGNING_OPTIONS TARGET_IPA_FILE APP_DESCRIPTOR SOURCE_FILES
```



参照 pfi を、pfi アプリケーションへのフルパスを含むように変更します。pfi アプリケーションは、Flash Professional CS5 インストールディレクトリの pfi/bin サブディレクトリにインストールされます。

作成する iPhone アプリケーションのタイプに対応する -target オプションを選択します。

- -target ipa-test - 開発 iPhone でアプリケーションのテスト用のバージョンを迅速にコンパイルするには、このオプションを選択します。
- -target ipa-debug — 開発 iPhone でアプリケーションのデバッグ用のバージョンをコンパイルするには、このオプションを選択します。このオプションを選択すると、デバッグセッションを使用して、iPhone アプリケーションから trace() 出力を受け取ることができます。

次のいずれかの -connect オプション (CONNECT\_OPTIONS) を含めて、デバッガを実行する開発コンピューターの IP アドレスを指定できます。

- -connect - アプリケーションは、アプリケーションのコンパイルに使用している開発コンピューター上のデバッグセッションに接続しようとします。
- -connect IP\_ADDRESS - アプリケーションは、指定した IP アドレスのコンピューター上のデバッグセッションに接続しようとします。次に、例を示します。  

```
-target ipa-debug -connect 192.0.32.10
```
- -connect HOST\_NAME - アプリケーションは、指定したホスト名のコンピューター上のデバッグセッションに接続しようとします。次に、例を示します。  

```
-target ipa-debug -connect bobroberts-mac.example.com
```

**注意：** -connect オプションは、Flash Professional CS5 に付属する Packager for iPhone Preview には含まれていません。Flash Professional CS5 でヘルプ/アップデートを選択して、Packager for iPhone を更新してください。

-connect オプションは必須ではありません。指定しない場合は、実行されるデバッグアプリケーションはホストされたデバッガーへの接続を試みません。

デバッグ接続が失敗した場合、アプリケーションはユーザーに、デバッグ用ホストマシンの IP アドレスを入力するように求めるダイアログを表示します。デバイスが Wi-Fi に接続されていない場合、接続が失敗する可能性があります。また、デバイスが接続されていても、デバッグ用ホストマシンのファイアウォールに阻まれる場合に、接続が失敗することがあります。

詳しくは、24 ページの「[iPhone アプリケーションのデバッグ](#)」を参照してください。

また、-renderingdiagnostics オプションを含めて、GPU レンダリング診断機能を有効にすることができます。詳しくは、24 ページの「[iPhone アプリケーションのデバッグ](#)」の、「[GPU レンダリング診断を使用したデバッグ](#)」を参照してください。

- -target ipa-ad-hoc — アドホックデプロイ用のアプリケーションを作成するには、このオプションを選択します。Apple iPhone デベロッパーセンターを参照してください。
- -target ipa-app-store — Apple App Store にデプロイする IPA ファイルの最終版を作成するには、このオプションを選択します。

PROFILE\_PATH を、アプリケーション用プロビジョニングプロファイルへのパスで置き換えます。プロビジョニングプロファイルについて詳しくは、4 ページの「[Apple から開発ファイル入手](#)」を参照してください。

SIGNING\_OPTIONS を、iPhone 開発用証明書およびパスワードを参照するように置き換えます。次のシンタックスを使用します。

```
-storetype pkcs12 -keystore P12_FILE_PATH -storepass PASSWORD
```

**P12\_FILE\_PATH** を、P12 証明書ファイルへのパスに置き換えます。**PASSWORD** を、証明書のパスワードに置き換えます（後述の例を参照）。P12 証明書ファイルについて詳しくは、7 ページの「[P12 ファイルへの開発用証明書の交換](#)」を参照してください。

APP\_DESCRIPTOR を、アプリケーション記述ファイルを参照するように置き換えます。

SOURCE\_FILES を、プロジェクトのメインの SWF ファイルを参照するように置き換え、その他に含めるアセットをその後ろに追加します。Flash CS5 のアプリケーション設定ダイアログボックスまたはカスタムアプリケーション記述ファイルで定義したすべてのアイコンファイルのパスを含めます。また、起動画面アートファイル (Default.png) も追加します。

次の例について考えます。

```
pfcli -package -target ipa-test -storetype pkcs12 -keystore  
"/Users/Jeff/iPhoneCerts/iPhoneDeveloper_Jeff.p12" -storepass dfb7VKL19 "HelloWorld.ipa" "HelloWorld-  
app.xml" "HelloWorld.swf" "Default.png" "icons/icon29.png" "icons/icon57.png" "icons/icon512.png"
```

この場合、以下のものを使用して HelloWorld.ipa ファイルがコンパイルされます。

- 証明書パスワード dfb7VKL19 を使用する特定の PKCS#12 証明書
- HelloWorld-app.xml アプリケーション記述ファイル
- ソースの HelloWorld.swf ファイル
- 特定の Default.png ファイルおよびアイコンファイル

pfcli アプリケーションは、アプリケーション記述ファイル、SWF ファイルおよびその他のアセットに基づいて、アプリケーションを IPA ファイルにコンパイルします。

Mac OS では、キーチェーンに格納されている証明書を使用できます。これを行うには、以下のオプションを pfcli コマンドに追加します。

```
-alias ALIAS_NAME -storetype KeychainStore -providerName Apple
```

ALIAS\_NAME を、使用する証明書のエイリアスに置き換えます。Mac のキーチェーンに格納されている証明書を指定する場合は、証明書ファイルの場所を指定する代わりに、このエイリアスを指定します。

## iPhone アプリケーションのインストール

開発用アプリケーションを iPhone にインストールするには、iPhone にプロビジョニングプロファイルを追加してから、アプリケーションをインストールします。

### iPhone へのプロビジョニングプロファイルの追加

iPhone にプロビジョニングプロファイルを追加するには：

- 1 iTunes で、ファイル/ライブラリに追加を選択します。次に、プロビジョニングプロファイル（ファイル名の拡張子が mobileprovision のファイル）を選択します。

iPhone がプロビジョニングプロファイルに追加されてい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は Apple iPhone Dev Center サイト (<http://developer.apple.com/iphone>) で管理できます。このサイトの iPhone デベロッパープログラムポータルセクションに移動してください。開発用証明書のインストール先のデバイス一覧を管理するには、「Devices」リンクをクリックします。プロビジョニングプロファイルを管理するには、「Provisioning」リンクをクリックします。

- 2 iPhone をコンピューターに接続して同期します。

プロビジョニングプロファイルの取得について詳しくは、4 ページの「[Apple から開発ファイル入手](#)」を参照してください。

### アプリケーションのインストール

開発用アプリケーションのインストール方法は、他の IPA ファイルをインストールする場合と同じです。

- 1 アプリケーションのバージョンが以前にインストールされている場合は、そのアプリケーションをデバイスおよび iTunes のアプリケーションリストから削除します。
- 2 以下のいずれかの方法で、iTunes にアプリケーションを追加します。
  - ファイルメニューから、「ライブラリに追加」コマンドを選択します。IPA ファイルを選択して、「開く」ボタンをクリックします。
  - IPA ファイルをダブルクリックします。
  - IPA ファイルを iTunes ライブラリにドラッグします。
- 3 iPhone を、コンピューターの USB ポートに接続します。
- 4 iTunes で、このデバイスの「アプリケーション」タブを確認し、インストールするアプリケーションのリストで該当するアプリケーションが選択されていることを確認します。
- 5 iPhone を同期します。

### アプリケーションのインストールに関する問題のトラブルシューティング

アプリケーションのインストール時に iTunes でエラーが表示された場合は、以下について確認してください。

- デバイス ID がプロビジョニングプロファイルに追加されていることを確認します。
- プロビジョニングプロファイルを確実にインストールするには、プロビジョニングプロファイルを iTunes にドラッグするか、ファイル/ライブラリに追加コマンドを使用します。

また、アプリケーションのアプリケーション ID が Apple アプリケーション ID と一致することを確認します。

- 例えば、Apple アプリケーション ID が `com.myDomain.*` の場合、アプリケーション記述ファイルのアプリケーション ID または iPhone 設定ダイアログボックスの「アプリケーション ID」は、`com.myDomain` で始まる必要があります (`com.myDomain.anythinghere` など)。
- Apple アプリケーション ID が `com.myDomain.myApp` の場合、アプリケーション記述ファイルまたは Flash Professional CS5 ユーザーインターフェイスのアプリケーション ID は、`com.myDomain.myApp` である必要があります。
- Apple アプリケーション ID が `*` の場合は、アプリケーション記述ファイルまたは Flash Professional CS5 ユーザーインターフェイスのアプリケーション ID には、任意の ID を使用できます。

アプリケーションのアプリケーション ID は、Flash Professional CS5 の iPhone 設定ダイアログボックスまたはアプリケーション記述ファイルで設定します。

## iPhone アプリケーションのデバッグ

開発コンピューターの ADL でアプリケーションを実行しながら、アプリケーションをデバッグできます。iPhone 上でアプリケーションをデバッグすることもできます。

iPhone 上でサポートされない AIR 機能の中には、(開発コンピューター上で) ADL を使用してアプリケーションをテストする際に利用できるものがあります。デスクトップ上でコンテンツをテストする際にこれらの違いについて気を付けてください。詳しくは、28 ページの「[モバイルデバイスでサポートされていない ActionScript 3.0 API](#)」を参照してください。

### 開発コンピューターでのアプリケーションのデバッグ

開発コンピューターで Flash Professional CS5 を使用してアプリケーションをデバッグするには：

- ❖ デバッグ/ムービーをデバッグ/ AIR Debug Launcher (モバイル) を使用を選択します。

コマンドラインから ADL を呼び出してアプリケーションをデバッグすることもできます。シンタックスは次の通りです。

```
adl -profile mobileDevice appDescriptorFile
```

appDescriptorFile をアプリケーション記述ファイルへのパスに置き換えます。

-profile mobileDevice オプションを必ず指定してください。

### iPhone でのアプリケーションのデバッグ

iPhone でアプリケーションをデバッグするには：

- 1 デバッグサポートを使用してアプリケーションをコンパイルします。
  - Flash Professional CS5 では、「デバイスのデバッグを目的としたクイックパブリッシュ」設定を使用してコンパイルします (21 ページの「[Flash Professional CS5 に付属する Packager for iPhone を使用した iPhone アプリケーションインストーラーファイルの作成](#)」を参照してください)。
  - PFI コマンドラインアプリケーションを使用する場合は、target ipa-debug オプションを指定してアプリケーションをコンパイルします (21 ページの「[コマンドラインからの iPhone アプリケーションインストーラーファイルの作成](#)」を参照してください)。
- 2 アプリケーションを iPhone にインストールします。
- 3 iPhone で Wi-Fi を有効にし、開発コンピューターと同じネットワークに接続します。
- 4 開発コンピューターでデバッグセッションを開始します。Flash Professional CS5 で、デバッグ/リモートデバッグセッションを開始/ ActionScript 3.0 を選択します。
- 5 iPhone でアプリケーションを実行します。

アプリケーションのデバッグ版で、開発コンピューターの IP アドレスを入力するように求められます。IP アドレスを入力して、「OK」ボタンをタップします。開発コンピューターの IP アドレスを入手する手順を次に示します。

- Mac OS で、Apple メニューのシステム環境設定を選択します。システム環境設定ウィンドウで、ネットワークアイコンをクリックします。ネットワーク環境設定ウィンドウに IP アドレスが表示されます。
- Windows で、コマンドラインセッションを開始し、ipconfig コマンドを実行します。

デバッグセッションで、アプリケーションからの trace() 出力が表示されます。

iPhone にインストールされているアプリケーションをデバッグする場合、Flash Professional CS5 では、すべてのデバッグ機能 (ブレークポイントの制御、コードのステップ実行、変数の監視など) をサポートします。

### GPU レンダリング診断を使用したデバッグ

GPU レンダリング診断機能を使用すると、アプリケーションによるハードウェアアクセラレーション (GPU レンダリングモードを使用するアプリケーション向け) の使用状況を確認することができます。この機能を使用するには、コマンドラインで PFI ツールを使用してアプリケーションをコンパイルします。この際に、次のように -renderingdiagnostics オプションを含めます。

```
pfi -package -renderingdiagnostics -target ipa-debug -connect ...
```

-renderingdiagnostics フラグは -package フラグの次に指定する必要があります。

GPU レンダリング診断機能は、すべての表示オブジェクトに対して次の色の長方形を表示します。

- 青 - 表示オブジェクトはビットマップではないか、ビットマップとしてキャッシュされておらず、レンダリングされません。

変更中ではない表示オブジェクトで繰り返し青が表示される場合、移動中の表示オブジェクトと交差している可能性があります。例えば、表示オブジェクトが、移動中の表示オブジェクトの背景である可能性があります。この表示オブジェクトをビットマップとしてキャッシュすることを検討してください。

キャッシュされていないオブジェクトで青が表示される場合は、GPU が適用できない効果をそのオブジェクトが使用している可能性があります。このような効果には、特定のブレンドモード、色変換、scrollRect プロパティおよびマスクがあります。

また、GPU にアップロードされる表示オブジェクトがメモリの制限容量を超える場合にも、アプリケーションは青を表示します。

アプリケーションは青の各長方形についてログメッセージを出力します。アプリケーションはこれらのメッセージを、他の trace() メッセージやデバッグ出力メッセージと共に出力します。

- 緑 - 表示オブジェクトはビットマップであるか、ビットマップとしてキャッシュされていて、GPU へ初回のアップロードが実行されます。

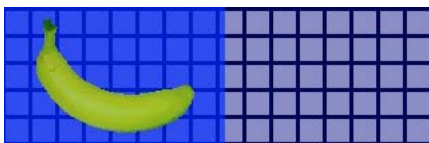
表示オブジェクトに対して緑が繰り返し表示される場合、アプリケーション内のコードはその表示オブジェクトを再作成しています。例えば、タイムラインが表示オブジェクトを作成するフレームに戻る場合に、このような状況が発生します。同じオブジェクトの再作成を防止するために、コンテンツを修正することを検討してください。

- 赤 - 表示オブジェクトはビットマップであるか、ビットマップとしてキャッシュされていて、GPU に再アップロードされます。

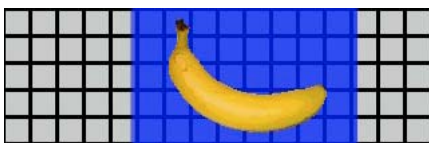
赤は、アプリケーションがビットマップ表現を再レンダリングする必要があるような状態で表示オブジェクトが変更されたタイミングで表示されます。例えば、cacheAsBitmapMatrix プロパティが設定されていない 2D オブジェクトは、拡大 / 縮小または回転時に再レンダリングされます。再レンダリングは、子表示オブジェクトが移動したり変更されたりした場合にも発生します。

色の付いた各長方形は、4 つの画面再描画サイクルの間に色付けの理由が発生しない限り、このサイクルの実行後に消えます。ただし、画面上で変更がない場合、この診断による色は変化しません。

例えば、ビットマップとしてキャッシュされないベクター背景の手前のビットマップ表示オブジェクト（バナナ）を考察します。バナナが最初にレンダリングされると、緑になります。背景が最初にレンダリングされると、青になります。

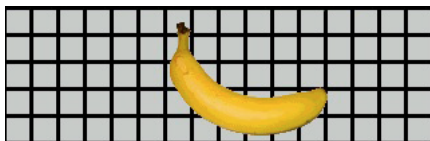


バナナが移動すると、CPU が背景を再レンダリングする必要があるため、背景上に青の陰影が表示されます。



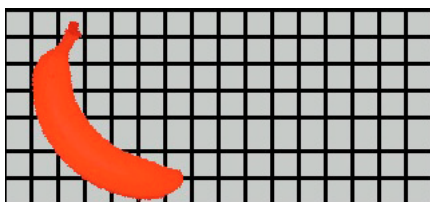
背景上の青の陰影は、GPU に送る必要のある、再描画された領域を表します。

ただし、背景がビットマップとしてキャッシュされている場合、バナナが移動しても、レンダリング診断機能は色を表示しません。



診断機能が色を表示しないのは、GPU が背景ビットマップを保持しているからです。GPU は CPU を使用せずに、バナナと背景を合成することができます。

バナナが `cacheAsBitmapMatrix` プロパティを設定していない 2D 表示オブジェクトであるとしてします。この表示オブジェクトが回転（または拡大 / 縮小）すると、レンダリング診断機能は赤を表示します。これは、アプリケーションが新しいバージョンの表示オブジェクトを GPU にアップロードする必要があることを示しています。



## App Store への iPhone アプリケーションの送信

App Store にアプリケーションを送信するには：

- 1 配布証明書とプロビジョニングプロファイルを iPhone Dev Center Web サイト (<http://developer.apple.com/iphone/>) から入手します。

配布証明書は「iPhone Developer: XXX.cer」という名前が付けられています。XXX はユーザー名です。

詳しくは、4 ページの「[Apple から開発ファイル入手](#)」を参照してください。

- 2 配布証明書を P12 ファイルに変換します。

詳しくは、7 ページの「[P12 ファイルへの開発用証明書の変換](#)」を参照してください。

- 3 P12 ファイルと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をコンパイルします。

配布証明書から作成した P12 ファイルを使用します。配布プロビジョニングプロファイルに関連付けられたアプリケーション ID を使用します。

詳しくは、21 ページの「[iPhone アプリケーションインストーラー \(IPA\) ファイルのコンパイル](#)」を参照してください。

- 4 iPhone Dev Center Web サイト (<http://developer.apple.com/iphone/>) にアプリケーションを送信します。

**重要：** Apple では、App Store へアプリケーションをアップロードするために、**Apple Application Loader** プログラムを使用することを要求しています。Apple は、Mac OS X 用 **Application Loader** のみを公開しています。Windows コンピューターを使用して iPhone 用 AIR アプリケーションを開発することはできますが、App Store にアプリケーションを送信するには、OS X (バージョン 10.5.3 以降) を実行するコンピューターにアクセスする必要があります。Application Loader プログラムは Apple iOS Dev Center から入手できます。

## 第3章：ActionScript 3.0 API によるモバイルデバイスのサポート

モバイル AIR アプリケーションを作成する際は、Adobe Flash Player 10.1 と AIR 2 デスクトップアプリケーションを開発する場合と同じ ActionScript 3.0 API を使用できます。ただし、例外と追加項目があります。

### モバイルデバイスでサポートされていない ActionScript 3.0 API

一部の ActionScript 3.0 API は、モバイルデバイスプロファイルで実行するアプリケーション（iPhone で実行するアプリケーションなど）ではサポートされていません。

同じ ActionScript コードを使用して、デスクトップやモバイルなど、各種のプロファイル用にアプリケーションを開発する場合は、API がサポートされているかどうかをテストするコードを使用します。例えば、NativeWindow クラスは iPhone アプリケーションではサポートされていません（iPhone アプリケーションでは、ネイティブウィンドウの使用も作成もできません）。ネイティブウィンドウをサポートするプロファイル（デスクトッププロファイルなど）でアプリケーションが実行されているかどうかをテストするには、NativeWindow.isSupported プロパティを確認します。

以下の表は、モバイルデバイスプロファイルでサポートされていない API を示します。また、この表には、API をサポートするプラットフォーム上でアプリケーションが実行されているかどうかを判断するために使用するプロパティも記載されています。

API	サポート状態のテスト
Accessibility	Capabilities.hasAccessibility
Camera	Camera.isSupported
DatagramSocket	DatagramSocket.isSupported
DNSResolver	DNSResolver.isSupported
DockIcon	NativeApplication.supportsDockIcon
DRMManager	DRMManager.isSupported
EncryptedLocalStore	EncryptedLocalStore.isSupported
HTMLLoader	HTMLLoader.isSupported
LocalConnection	LocalConnection.isSupported
Microphone	Microphone.isSupported
NativeApplication.exit()	—
NativeApplication.menu	NativeApplication.supportsMenu
NativeApplication.isSetAsDefaultApplication()	NativeApplication.supportsDefaultApplication
NativeApplication.startAtLogin	NativeApplication.supportsStartAtLogin
NativeMenu	NativeMenu.isSupported

API	サポート状態のテスト
NativeProcess	NativeProcess.isSupported
NativeWindow	NativeWindow.isSupported
NativeWindow.notifyUser()	NativeWindow.supportsNotification
NetworkInfo	NetworkInfo.isSupported
PDF サポート	HTMLLoader.pdfCapability
PrintJob	PrintJob.isSupported
SecureSocket	SecureSocket.isSupported
ServerSocket	ServerSocket.isSupported
シェーダー	—
ShaderFilter	—
StorageVolumeInfo	StorageVolumeInfo.isSupported
XMLSignatureValidator	XMLSignatureValidator.isSupported

モバイルデバイスプロファイル用の HTML ベースおよび JavaScript ベースの AIR アプリケーションを記述することはできません。

次に示す一部の ActionScript 3.0 のクラスは、部分的にしかサポートされません。

#### ファイル

iPhone アプリケーションは、アプリケーションディレクトリおよびアプリケーションストレージディレクトリのみアクセスできます。File.createTempFile() および File.createTempDirectory() を呼び出すこともできます。操作を呼び出して別のディレクトリ (FileStream の read または write メソッドなど) にアクセスすると、IOError の例外が発生します。

iPhone アプリケーションでは、File.browseForOpen() メソッドで提供されるような、ネイティブファイルブラウザーダイアログボックスはサポートされていません。

#### Loader

iPhone アプリケーションでは、Loader.load() メソッドを使用できません。Loader.load() メソッドを使用してロードされた SWF コンテンツ内の ActionScript コードを実行することができません。ただし、SWF ファイル内のアセット (ライブラリ内のムービークリップ、イメージ、フォント、音声など) は使用できます。また、イメージファイルをロードするために Loader.load() を使用することもできます。

#### ビデオ

iPhone 上の AIR アプリケーションでサポートされるのは、Sorensen ビデオおよび ON2 VP6 ビデオだけです。

navigateToURL メソッドを使用して、アプリケーションの外で H.264 ビデオを開くことができます。request パラメーターとして、目的のビデオを示す URL と共に URLRequest オブジェクトを渡します。指定されたビデオが、iPhone のビデオプレーヤーで起動されます。

#### テキストフィールド

iPhone 上のテキストフィールドのフォントおよびその他の設定には制限事項があります。38 ページの「[フォントとテキスト入力](#)」を参照してください。



### ADL を使用したサポート対象外の API およびデバッグ

iPhone 上でサポートされない AIR 機能の中には、(開発コンピューター上で) ADL を使用してアプリケーションをテストする際に利用できるものがあります。ADL を使用してコンテンツをテストする際にこれらの違いについて気を付けてください。

この機能には、Speex (オーディオコーデック)、H.264/AVC (ビデオコーデック) および AAC (オーディオコーデック) が含まれています。これらのコーデックは iPhone で実行する AIR アプリケーションでは利用できません。ただし、デスクトップ上では正常に動作します。

Windows の ADL では、ユーザー補助機能および文字読み上げ機能が動作します。しかし、これらの API は iPhone ではサポートされません。

RTMPE プロトコルはデスクトップ上の ADL から使用する場合には正常に動作します。しかし、RTMPE プロトコルを使用して接続を試みる NetConnection は、iPhone では成功しません。

Loader クラスは、コンテンツが ADL で実行される場合は、特に制限なく動作します。しかし、iPhone 上で実行される場合は、ActionScript バイトコードを含む SWF コンテンツをロードしようとすると、エラーメッセージが出力されます。

Shader インスタンスは ADL で実行できます。しかし、iPhone 上では、ピクセルベンダーバイトコードは解釈されず、シェーダーによるグラフィカルな効果が表れません。

詳しくは、24 ページの「[iPhone アプリケーションのデバッグ](#)」を参照してください。

## モバイル AIR アプリケーション固有の ActionScript API

以下に示す API は、モバイルデバイス上の AIR アプリケーションでのみ利用できます。現時点では、これらの API は Flash Player またはデスクトップバージョンの AIR では機能しません。

### Screen orientation (画面の向き) API

Screen orientation (画面の向き) API を使用すると、ステージおよび iPhone の向きに対応することができます。

- Stage.autoOrients — デバイスが回転したときに、ステージの向きが自動的に変更されるようにアプリケーションが設定されているかどうかを示します。Flash Professional CS5 の iPhone 設定ダイアログボックスで「自動回転を有効にする」オプションが選択されている場合、このプロパティには true が設定されます (アプリケーション記述ファイル内で、autoOrients エレメントに true を設定することもできます)。15 ページの「[iPhone アプリケーション設定](#)」を参照してください。向きの自動回転を解除するには、Stage オブジェクトの orientationChanging イベントリスナーを追加します。このイベントオブジェクトの preventDefault() メソッドを呼び出すと、自動回転が解除されます。

自動回転の使用時に最適な結果を得るために、Stage の align プロパティを次のように設定します。

```
stage.align = StageAlign.TOP_LEFT;  
stage.scaleMode = StageScaleMode.NO_SCALE;
```

- Stage.deviceOrientation — デバイスの物理的な向きです。StageOrientation クラスは、このプロパティの値を定義します。
- Stage.orientation — ステージの現在の向きです。StageOrientation クラスは、このプロパティの値を定義します。
- Stage.supportsOrientationChange — iPhone では true、AIR アプリケーションでは false に設定されます。
- Stage.setOrientation() — ステージの向きを設定します。このメソッドには 1 つのパラメーター (変更後のステージの向きを定義する文字列) が用意されています。StageOrientation クラスの定数によって、パラメーターに設定可能な値を定義します。
- StageOrientation — ステージの向きを示す値を定義します。例えば、StageOrientation.ROTATED\_RIGHT はデバイスのデフォルトの向きに対して右方向に回転したステージを表します。

- **StageOrientationEvent** — 画面の向きが変化した際に Stage オブジェクトが送信するイベントを定義します。このイベントは、ユーザーが iPhone を回転させたときに発生します。イベントには、2 種類あります。デバイスが回転されると、Stage オブジェクトは **orientationChanging** イベントを送信します。ステージの向きが変更されないようにするには、**orientationChanging** イベントオブジェクトの **preventDefault()** メソッドを呼び出します。ステージの向きの変更が完了すると、Stage オブジェクトは **orientationChange** イベントを送信します。

現時点で Screen orientation (画面の向き) API が有効なのは、モバイルデバイス上の AIR アプリケーションのみです。モバイル AIR アプリケーションとデスクトップ AIR アプリケーションでソースコードを共有する場合は、**Stage.supportsOrientationChange** プロパティを使用して、API がサポートされているかどうかを確認してください。

次の例では、ユーザーがデバイスを回転したときの対応方法を示します。

```
stage.addEventListener(StageOrientationEvent.ORIENTATION_CHANGE,
    onOrientationChange);

function onOrientationChange(event:StageOrientationEvent):void
{
    switch (event.afterOrientation) {
        case StageOrientation.DEFAULT:
            // re-orient display objects based on
            // the default (right-side up) orientation.
            break;
        case StageOrientation.ROTATED_RIGHT:
            // Re-orient display objects based on
            // right-hand orientation.
            break;
        case StageOrientation.ROTATED_LEFT:
            // Re-orient display objects based on
            // left-hand orientation.
            break;
        case StageOrientation.UPSIDE_DOWN:
            // Re-orient display objects based on
            // upside-down orientation.
            break;
    }
}
```

この例では、個々のステージの向きに対して、機能するコードの代わりにコメントが記述されています。

**Stage** オブジェクトの **setOrientation()** メソッドを呼び出すことで、ステージの向きを変更できます。向きの設定の非同期操作です。向きの変更がいつ完了するかを確認するには、**orientationChange** イベントを監視します。次のコードでは、ステージを右方向に設定する方法を示します。

```
stage.addEventListener(StageOrientationEvent.ORIENTATION_CHANGE,
    onOrientationChange);
stage.setOrientation(StageOrientation.ROTATED_RIGHT);

function onOrientationChange(event:StageOrientationEvent):void
{
    // Code to handle the new Stage orientation
}
```

ステージが回転すると、ステージのサイズが変更され、**Stage** オブジェクトは **resize** イベントを送出します。**resize** イベントに応じてステージ上の表示オブジェクトのサイズと位置を変更できます。

### NativeApplication.systemIdleMode および SystemIdleMode

**NativeApplication.systemIdleMode** プロパティを使用すると、iPhone がアイドルモードに移行するのを防ぐことができます。デフォルトでは、タッチスクリーンに一定期間接触しないと、iPhone はアイドルモードに移行します。アイドルモードでは画面が暗くなります。また、iPhone はロックモードに移行します。このプロパティは、以下に示す 2 つの値のいずれかに設定できます。

- **SystemIdleMode.NORMAL** - iPhone は通常の動作に従ってアイドルモードに移行します。

- `SystemIdleMode.KEEP_AWAKE` - アプリケーションでは iPhone がアイドルモードに移行しないように試みます。

この機能はモバイルデバイスでのみサポートされています。デスクトップオペレーティングシステムで実行する AIR アプリケーションではサポートされていません。デスクトップ上で動作するアプリケーションでは、`NativeApplication.systemIdleMode` プロパティを設定しても効果はありません。

以下のコードは、iPhone のアイドルモードを無効化する方法を示しています。

```
NativeApplication.nativeApplication.systemIdleMode = SystemIdleMode.KEEP_AWAKE;
```

## CameraRoll

`CameraRoll` クラスを使用すると、イメージを iPhone カメラロールに追加できます。`addBitmapData()` メソッドは、イメージを iPhone カメラロールに追加します。このメソッドには `bitmapData` というパラメーターが用意されています。このパラメーターは `BitmapData` オブジェクトで、カメラロールに追加するイメージが含まれています。

`CameraRoll` 機能はモバイルデバイスでのみサポートされています。デスクトップオペレーティングシステムで実行する AIR アプリケーションではサポートされていません。アプリケーションで `CameraRoll` 機能がサポートされているかどうかを実行時にチェックするには、`CameraRoll.supportsAddBitmapData` 静的プロパティを確認します。

`addBitmapData()` メソッドが呼び出されると、`CameraRoll` オブジェクトは以下に示す 2 つのイベントのいずれかを送出します。

- `complete` - 操作は正常に完了しました。
- `error` - エラーが発生しました。例えば、iPhone にイメージを格納するための十分な空き領域がない場合などです。

以下のコードは、ステージのイメージ（スクリーンキャプチャ）をカメラロールに追加します。

```
if (CameraRoll.supportsAddBitmapData)
{
    var cameraRoll:CameraRoll = new CameraRoll();
    cameraRoll.addEventListener(ErrorEvent.ERROR, onCrError);
    cameraRoll.addEventListener(Event.COMPLETE, onCrComplete);
    var bitmapData:BitmapData = new BitmapData(stage.stageWidth, stage.stageHeight);
    bitmapData.draw(stage);
    cameraRoll.addBitmapData(bitmapData);
}
else
{
    trace("not supported.");
}

function onCrError(event:ErrorEvent):void
{
    // Notify user.
}

function onCrComplete(event:Event):void
{
    // Notify user.
}
```

## DisplayObject.cacheAsBitmapMatrix

`cacheAsBitmapMatrix` プロパティは、`cacheAsBitmap` が `true` に設定されている場合の表示オブジェクトのレンダリング方法を定義する `Matrix` オブジェクトです。アプリケーションでは、表示オブジェクトのビットマップバージョンをレンダリングする際に、このマトリックスを変換マトリックスとして使用します。

cacheAsBitmapMatrix が設定されていると、アプリケーションでは、表示マトリックスではなくこのマトリックスを使用してレンダリングされたビットマップイメージのキャッシュを保持します（表示マトリックスとは、表示オブジェクトの transform.concatenatedMatrix の値のことです）。このマトリックスが表示マトリックスと一致しない場合は、必要に応じてビットマップを拡大 / 縮小および回転します。

cacheAsBitmapMatrix が設定されている表示オブジェクトは、cacheAsBitmapMatrix の値が変更された場合にのみレンダリングされます。ビットマップは表示マトリックスに応じて拡大 / 縮小または回転されます。

CPU ベースと GPU ベースのレンダリングはいずれも cacheAsBitmapMatrix プロパティを使用することによる効果を得られます。ただし、通常は GPU レンダリングの方が高い効果を得られます。

**注意：**ハードウェアアクセラレーションを使用するには、Flash Professional CS5 の iPhone 設定ダイアログボックスにある「一般」タブで、「レンダリング」を「GPU」に設定します。または、アプリケーション記述ファイル内で、renderMode プロパティを gpu に設定します。

例えば、以下のコードでは、表示オブジェクトの未変換のビットマップ表現を使用します。

```
matrix:Matrix = new Matrix(); // creates an identity matrix
mySprite.cacheAsBitmapMatrix = matrix;
mySprite.cacheAsBitmap = true;
```

以下のコードでは、現在のレンダリングと一致するビットマップ表現を使用します。

```
mySprite.cacheAsBitmapMatrix = mySprite.transform.concatenatedMatrix;
mySprite.cacheAsBitmap = true;
```

通常は、単位マトリックス (new Matrix()) または transform.concatenatedMatrix で十分ですが、別のマトリックス (縮小マトリックスなど) を使用すれば、異なるビットマップを GPU にアップロードできます。例えば、以下の例では cacheAsBitmapMatrix マトリックスを適用して、x 軸および y 軸方向に 0.5 倍に縮小します。GPU が使用するビットマップは小さくなりますが、GPU ではビットマップのサイズを表示オブジェクトの transform.matrix プロパティに合わせて調整します。

```
matrix:Matrix = new Matrix(); // creates an identity matrix
matrix.scale(0.5, 0.5); // scales the matrix
mySprite.cacheAsBitmapMatrix = matrix;
mySprite.cacheAsBitmap = true;
```

通常は、アプリケーションで表示されるサイズに合わせて表示オブジェクトを変換するマトリックスを選択します。例えば、スプライトについて、サイズを半分に縮小したビットマップバージョンをアプリケーションで表示する場合は、半分の縮小率のマトリックスを使用します。スプライトを現在のサイズよりも拡大してアプリケーションに表示する場合は、該当する拡大率のマトリックスを使用します。

cacheAsBitmapMatrix プロパティを設定する場合、表示オブジェクトのサイズには、実質的な制限があります。その制限は 1,020 x 1,020 ピクセルです。cacheAsBitmapMatrix プロパティを設定する場合、すべての表示オブジェクトの総ピクセル数には、実質的な制限があります。その制限は約 400 万ピクセルです。

cacheAsBitmapMatrix およびハードウェアアクセラレーションの使用に際しては、様々な考慮事項があります。このプロパティを設定する必要があるかどうか、表示オブジェクトごとに把握しておくことが重要です。このプロパティを使用する上での重要な情報については、35 ページの「ハードウェアアクセラレーション」を参照してください。

GPU レンダリング診断機能を使用して、アプリケーションのデバッグビルドで GPU 使用状況を診断できます。詳しくは、24 ページの「iPhone アプリケーションのデバッグ」を参照してください。

### ネットワークに関する注意事項

navigateToURL() 関数で次の URL スキームを使用すると、ドキュメントが外部のアプリケーションで開きます。

URL スキーム	nativeToURL() 呼び出しの結果	例
mailto:	メールアプリケーションで新規のメッセージを開きます。	<pre>str = "mailto:test@example.com"; var urlReq:URLRequest = new URLRequest(str); navigateToURL(urlReq);</pre>
sms:	テキストメッセージアプリケーションでメッセージを開きます。	<pre>str = "sms:1-415-555-1212"; var urlReq:URLRequest = new URLRequest(str); navigateToURL(urlReq);</pre>
tel:	ユーザーの承認後に電話で電話番号をダイヤルします。	<pre>str = "tel:1-415-555-1212"; var urlReq:URLRequest = new URLRequest(str); navigateToURL(urlReq);</pre>

https リクエストなどの保護されたトランザクションを実行中に、iPhone アプリケーションが、インストールされた自己署名ルート証明書を基にサーバー認証を行う場合があります。サーバーはリーフ証明書だけでなく、ルート証明書に関連付けられたすべての中間証明書を送信する必要があります。

## モバイルアプリケーションの開発者に特に有用な ActionScript 3.0 API

以下の ActionScript 3.0 API では、モバイルデバイスに役立つ機能を定義します。

### Accelerometer (加速度センサー) API

次のクラスを使用すると、デバイスの加速度センサーからイベントを取得できます。

- Accelerometer
- AccelerometerEvent

詳しくは、「[加速度センサー入力](#)」を参照してください。

### Geolocation (ジオロケーション) API

次のクラスを使用すると、デバイスの位置センサーからイベントを取得できます。

- Geolocation
- GeolocationEvent

詳しくは、「[ジオロケーション](#)」を参照してください。

### タッチ、マルチタッチ、ジェスチャー API

次のクラスを使用すると、タッチやジェスチャーのイベントを受信できます。

- GestureEvent
- GesturePhase
- MultiTouch
- MultitouchInputMode
- TouchEvent
- TransformGestureEvent

詳しくは、「[タッチ入力](#)、[マルチタッチ入力](#)および[ジェスチャー入力](#)」を参照してください。

## 第4章：iPhone アプリケーションのデザイン上の考慮事項

iPhone の処理速度と画面サイズは、特殊なデザインやコーディングを考慮する上で役立ちます。ただし、デザイン上の考慮事項の多くは、すべてのアプリケーションまたはモバイルアプリケーションに共通しています。

アプリケーションの最適化について詳しくは、「[Flash Platform のパフォーマンスの最適化](#)」を参照してください。このドキュメントには、一般的に、モバイルコンテンツ、Flash Player コンテンツ、AIR コンテンツおよび ActionScript ベースのコンテンツのパフォーマンスを最適化するための推奨事項が数多く記載されています。これらの推奨事項の多くは、iPhone 用の AIR アプリケーションにも適用されます。

**重要：**デザイン上の考慮事項および最適化テクニックの多くは、iPhone で快適に動作するアプリケーションを開発する上で重要です。

### ハードウェアアクセラレーション

OpenGL ES 1.1 ハードウェアアクセラレーションを使用すると、一部のアプリケーションでグラフィックのパフォーマンスを向上させることができます。表示オブジェクトがアニメーション化される、ゲームなどのアプリケーションでは、ハードウェアアクセラレーションによって利点が得られます。ハードウェアアクセラレーションを使用するアプリケーションは、CPU から iPhone の GPU に一部のグラフィック処理の負荷を分散させることができ、これによりパフォーマンスが著しく向上します。

GPU を使用するアプリケーションをデザインする際は、コンテンツが効果的に GPU アクセラレーションの対象となるようにルールに従うことが重要です。

ハードウェアアクセラレーションを使用するには、Flash Professional CS5 の iPhone 設定ダイアログボックスにある「一般」タブで、「レンダリング」を「GPU」に設定します。アプリケーション記述ファイル内で、renderMode プロパティを gpu に設定することもできます。

```
<initialWindow>
  <renderMode>gpu</renderMode>
  ...
```

15 ページの「[Flash Professional CS5 における iPhone アプリケーションプロパティの設定](#)」または 17 ページの「[アプリケーション記述ファイルにおける iPhone アプリケーションプロパティの設定](#)」を参照してください。

コンテンツが頻繁に更新されない場合に、ハードウェアアクセラレーションによって迅速にレンダリングできる表示オブジェクトには次の 4 つのクラスがあります。

- Bitmap オブジェクト
- cacheAsBitmap プロパティが true に設定され、オプションとして cacheAsBitmapMatrix プロパティが設定されている 2D 表示オブジェクト（以降の説明を参照）
- 3D 表示オブジェクト（つまり z プロパティが設定されたオブジェクト）
- 単色の矩形で塗りつぶされ、エッジが画面上のピクセルに沿って並んでいる表示オブジェクト

ベクターベースのオブジェクトは、他のスプライトがオブジェクトの上または下でアニメーション化されるたびにレンダリングされます。したがって、アニメーションのバックドロップまたは前景の役割を果たすオブジェクトも、これらのいずれかのカテゴリに属する必要があります。

cacheAsBitmap が true に設定されている表示オブジェクトの場合、cacheAsBitmapMatrix を設定すると、GPU はマトリックス変換結果のビットマップを使用することになります。GPU は、オブジェクトが回転または拡大 / 縮小されている場合でも、ビットマップ表現を使用します。GPU では、CPU がベクターレンダリングされたオブジェクトを再描画するよりもはるかに高速にこのビットマップを合成してアニメーション化できます。

cacheAsBitmap を true に設定するのみの場合、表示オブジェクト（およびその子）がキャッシュされるようになります。このような表示オブジェクトは、新しい領域が表示される時、または結合されたグラフィック全体が平行移動される時に再描画されません。

表示オブジェクトの cacheAsBitmapMatrix プロパティを設定すると、アプリケーションは、非表示の場合でも表示オブジェクトの表現を作成できます。アプリケーションは、次のフレームの開始時に表示オブジェクトのキャッシュを作成します。その後、表示オブジェクトがステージに追加されると、直ちにその表示オブジェクトをレンダリングします。オブジェクトの回転または拡大 / 縮小も迅速に行うことができます。回転または拡大 / 縮小するオブジェクトの場合、cacheAsBitmap プロパティを設定するときは必ず cacheAsBitmapMatrix プロパティも設定してください。

また、アプリケーションは、ビットマップとしてキャッシュされたオブジェクトのアルファ変換を高速で実行することもできます。ただし、ハードウェアアクセラレーションによるアルファ変換では、0 ~ 1.0 の alpha 値のみがサポートされます。これは、0 ~ 256 の colorTransform.alphaMultiplier 設定に相当します。

テキストフィールドなどの頻繁に更新されるオブジェクトに対しては、cacheAsBitmap プロパティを true に設定しないでください。

頻繁に変更されるグラフィカルコンテンツのある表示オブジェクトは、一般的に GPU レンダリングには向いていません。これは特に、GPU の性能が低い初期のデバイスで当てはまります。グラフィックを GPU にアップロードするオーバーヘッドを考慮すると、CPU レンダリングの方が適切である可能性があります。

親との相対位置を維持して移動する子表示オブジェクトを含む表示オブジェクトを再構成してください。子表示オブジェクトが親の兄弟となるように変更してください。これによって、各オブジェクトが独自のビットマップ表現を持つようになります。また、各表示オブジェクトは、新しいグラフィックを GPU にアップロードする必要なしに、他のオブジェクトとの相対位置を維持して移動できるようになります。

cacheAsBitmap プロパティを、子表示オブジェクトがアニメーション化されない最大レベルで、true に設定します。つまり、可動パーツを含まない表示オブジェクトコンテナ用に設定します。子表示オブジェクトには設定しないでください。アニメーション化するその他の表示オブジェクトを含むスプライトには設定しません。

表示オブジェクトの z プロパティを設定すると、アプリケーションは常に、キャッシュされたビットマップ表現を使用します。また、表示オブジェクトの z プロパティが設定された後では、オブジェクトを回転または拡大 / 縮小する場合でも、キャッシュされたビットマップ表現を使用します。z プロパティが設定されている表示オブジェクトでは、cacheAsBitmapMatrix プロパティが使用されません。同じルールが、rotationX、rotationY、rotationZ、transform.matrix3D プロパティなどの 3D 表示オブジェクトのプロパティを設定する際に適用されます。

ハードウェアアクセラレーションを使用するコンテンツを含む表示オブジェクトコンテナについては、scrollRect プロパティおよび mask プロパティを設定しないでください。これらのプロパティを設定すると、表示オブジェクトコンテナおよびその子オブジェクトのハードウェアアクセラレーションが無効になります。mask プロパティを設定する代わりに、マスクされている表示オブジェクトの上にマスク表示オブジェクトを重ね合わせてください。

ハードウェアアクセラレーションを利用できる表示オブジェクトのサイズには制限があります。古いデバイスでは、この制限は幅、高さともに 1024 ピクセル以下です。新しいデバイスでは、この制限は 2048 ピクセル以下です。GPU レンダリング診断ツールを使用して、デバイス上でのパフォーマンスをテストすることができます。

GPU はビットマップイメージを保存するために、iPhone RAM も使用します。少なくとも、ビットマップイメージに必要な容量と同じメモリ容量を使用します。

GPU は各ビットマップイメージの容量に対して 2 の累乗のメモリを割り当てます。例えば、GPU は 512 x 1024 や 8 x 32 のサイズのメモリを予約することができます。このため、9 x 15 ピクセルのイメージは、16 x 16 ピクセルのイメージに相当するメモリ容量を消費します。キャッシュされた表示オブジェクトでは、各辺で 2 の累乗に近い（ただし超過しない）サイズを使用するとよいでしょう。例えば、33 x 17 ピクセルの表示オブジェクトよりも、32 x 16 ピクセルの表示オブジェクトを使用する方が効率的です。

他のプラットフォーム（デスクトップなど）向けにサイズ設定されたアセットを縮小する目的で、サイズ変更された Stage を使用しないでください。代わりに、`cacheAsBitmapMatrix` プロパティを使用するか、アセットのサイズを変更してから iPhone にパブリッシュしてください。3D オブジェクトは表面イメージをキャッシュする際に `cacheAsBitmapMatrix` プロパティを無視します。このため、表示オブジェクトが 3D 表面にレンダリングされる場合は、サイズを変更してからパブリッシュする方が適切です。

ハードウェアアクセラレーションの利点と RAM の使用にはトレードオフがあります。メモリの空きがなくなると、iPhone OS は他の実行中のネイティブ iPhone アプリケーションにメモリを解放するよう通知します。これらのアプリケーションがこの通知を処理しメモリを解放するように動作するため、CPU サイクルにおいて自分のアプリケーションと競合する可能性があります。これにより、アプリケーションのパフォーマンスが一時的に低下する場合があります。古いデバイスでアプリケーションをテストするようにしてください。古いデバイスでは、実行中のプロセスが利用できるメモリの容量が大幅に少なくなります。

iPhone でアプリケーションをデバッグする際に、GPU レンダリング診断機能を有効にできます。この機能によって、アプリケーションによる GPU レンダリングの使用状況を確認できます。詳しくは、24 ページの「[iPhone アプリケーションのデバッグ](#)」の、「[GPU レンダリング診断を使用したデバッグ](#)」を参照してください。

`cacheAsBitmapMatrix` プロパティの使用については、30 ページの「[モバイル AIR アプリケーション固有の ActionScript API](#)」の「`DisplayObject.cacheAsBitmapMatrix`」を参照してください。

## 表示オブジェクトのパフォーマンスを向上させるその他の方法

ハードウェアアクセラレーションを使用すると、表示オブジェクトの一部のクラスでグラフィックのパフォーマンスを向上させることができます。グラフィックのパフォーマンスを最大化する方法のヒントを、いくつか示します。

- ステージに表示するアイテムの数を制限してください。アイテムはそれぞれ、レンダリングおよび周囲の他のアイテムとの合成に時間がかかります。

表示オブジェクトが不要になった場合には、`visible` プロパティを `false` に設定するか、または (`removeChild()` を使用して) ステージから削除します。`alpha` プロパティを 0 に設定するだけの処理は行わないでください。

- 大半のブレンドモードの使用は避けてください。特に、レイヤーブレンドモードの使用は避けてください。可能な限り、標準ブレンドモードを使用します。
- 表示オブジェクトフィルターを使用すると、コンピューター処理上の負荷が高くなります。このフィルターの使用は控えめにしてください。例えば、初期画面で少数のフィルターを使用する程度なら問題ありません。ただし、多くのオブジェクトまたはアニメーション中のオブジェクトでフィルターを使用することや、高いフレームレートを使用する必要があるときにフィルターを使用することは避けてください。
- モーフィングシェイプは避けてください。
- クリッピングの使用は避けてください。
- 可能であれば、`Graphic.beginBitmapFill()` メソッドを呼び出すときは、`repeat` パラメーターを `false` に設定します。
- 描画を誇張しないでください。背景色は背景として使用し、大きいシェイプは互いに重ね合わせないでください。描画時には、ピクセルごとに負荷がかかります。これは特に、ハードウェアアクセラレーションの対象ではない表示オブジェクトに当てはまります。



- 細長くとがった部分や自己交差しているエッジを含むシェイプ、エッジ部分が細密になっているシェイプの作成は避けてください。これらのシェイプは、スムーズなエッジを持つ表示オブジェクトよりもレンダリングに時間がかかります。これは特に、ハードウェアアクセラレーションの対象ではない表示オブジェクトに当てはまります。
- ビットマップは、 $2^n \times 2^m$  ビットに近い、それ未満のサイズにしてください。サイズは 2 の累乗とする必要はありませんが、2 の累乗を超えない範囲で、それに近いものである必要があります。例えば、 $31 \times 15$  ピクセルのイメージは、 $33 \times 17$  ピクセルのイメージよりも高速にレンダリングされます（31 と 15 は、2 の累乗である 32 と 16 の直前の整数です）。また、このようなイメージは、メモリを効率よく使用します。
- 表示オブジェクトのサイズを  $1024 \times 1024$  ピクセル（新しいデバイスの場合は  $2048 \times 2048$ ）に制限してください。

## 情報密度

モバイルデバイスの画面をデスクトップの画面と比較すると、物理的なサイズは小さくても、ピクセル密度は高くなっています。テキストは鮮明なほうが見栄えが良いものの、字形には判読可能な最小の物理的サイズを指定する必要があります。

モバイルデバイスは、多くの場合、移動中や、照明条件が悪い中で使用されます。オンスクリーンでどれだけ多くの情報を現実的に読みやすく表示できるかを検討します。モバイルデバイスに表示できる情報量は、デスクトップで同じピクセルサイズの画面に表示するよりも少なくなる場合があります。

階層化されたテキスト書式を使用して重要な情報を強調してください。フォントサイズ、線の太さ、配置および間隔を使用して、ユーザーインターフェイスの各要素の相対的な重要度を表現します。階層のレベルごとにキューを 1 つ以上使用できます。これらのキューは、アプリケーション全体に適用してください。キューには、間隔（インデント、行間、配置）またはグラフィック（書体のサイズ、スタイル、色）を使用できます。重複キューを適用することは、階層を明確に表示するための効果的な方法です。ただし、グループの各レベルに 3 つを超えるキューを使用しないようにしてください。

必要なラベルや説明テキストは、簡単にしてください。例えば、テキストフィールドでサンプル入力を使用して内容を示せば、別途ラベルを作成する必要はありません。

## フォントとテキスト入力

最良の外観を保持するには、デバイスフォントを使用してください。例えば、以下の各フォントは iPhone のデバイスフォントです。

- Serif : Times New Roman、Georgia および `_serif`
- Sans-serif : Helvetica、Arial、Verdana、Trebuchet、Tahoma および `_sans`
- 固定幅 : Courier New、Courier および `_typewriter`

14 ピクセル以上のフォントを使用します。

編集可能なテキストフィールドには、デバイスフォントを使用します。デバイスフォントは、テキストフィールド内でも埋め込みフォントに比べて速くレンダリングされます。

入力テキストフィールドに下線付きのテキストは使用しないでください。また、テキストフィールドの行揃えを設定しないでください。iPhone の入力テキストフィールドは、左揃え（デフォルト）のみをサポートしています。

Flash Professional CS5 内のテキストフィールドで TLF テキスト設定を使用する場合は、ActionScript 3.0 設定のデフォルトリンケージで、ランタイム共有ライブラリを無効にします。このようにしないと、アプリケーションはランタイム共有ライブラリの SWF ファイルを使用しようとするので、iPhone で動作しなくなります。

- 1 ファイル/パブリッシュ設定を選択します。
- 2 パブリッシュ設定ダイアログボックスで、「Flash」タブをクリックします。

3 「スクリプト」ドロップダウンリスト（「ActionScript 3.0」を選択）の右にある「設定」ボタンをクリックします。

4 「ライブラリパス」タブをクリックします。

5 「デフォルトリンケージ」ドロップダウンリストで、「コードにマージ」を選択します。

入力テキストフィールドの代わりとなる機能を実装することを検討してください。例えば、ユーザー入力が数値の場合、テキストフィールドは必要ありません。値を増減する 2 つのボタンを代わりに使うことができます。

仮想キーボードが占めるスペースに注意してください。仮想キーボードが有効になっている場合（ユーザーがテキストフィールド内をタップした場合など）には、アプリケーションによってステージの位置が調整されます。自動の位置調整機能により、選択された入力テキストフィールドは必ず見えるように調整されます。

- ステージの最上部にあるテキストフィールドは、画面上に表示されるステージ領域の最上部に移動します（画面上に表示されるステージ領域は、仮想キーボードを表示するために縮小されています）。
- ステージの最下部にあるテキストフィールドは、新しいステージ領域でも最下部の位置に留まります。
- ステージ上の他の位置にあるテキストフィールドは、ステージの垂直方向の中心に移動されます。

ユーザーが編集のためにテキストフィールドをクリックすると（仮想キーボードが表示され）、TextField オブジェクトが focusIn イベントを送出します。このイベントに、テキストフィールドの位置を変更するイベントリスナーを追加できます。

単一行テキストフィールドには、テキスト編集時に「消去」ボタンが（テキストの右側に）含まれます。しかし、テキストフィールドが小さすぎる場合は、「消去」ボタンは表示されません。

単一行テキストフィールドでテキストを編集後にキーボードの Done キーをタップすると、仮想キーボードが表示されなくなります。

複数行テキストフィールドを編集後にテキストフィールドの外側をタップすると、仮想キーボードが表示されなくなります。これによって、テキストフィールドからフォーカスが除去されます。仮想キーボードが表示される際には、テキストフィールドの外側の領域が含まれるようにデザインしてください。テキストフィールドが大きすぎると、他の領域が見えなくなることがあります。

Flash Professional CS5 コンポーネントの中には、テキストフィールドからフォーカスを除去しないものがあります。これらのコンポーネントは、このようなフォーカスの動作が望まれるようなデスクトップマシン上での使用を目的としてデザインされています。このようなコンポーネントの例として、TextArea コンポーネントがあります。TextArea にフォーカスがある（編集中の）ときに、他の表示オブジェクトをクリックしてもフォーカスは除去されません。これ以外の Flash Professional CS5 コンポーネントでも、ステージ上に置くことで編集中のテキストフィールドからフォーカスが変更されなくなることがあります。

キーボードイベントに依存しないでください。例えば、Web 用にデザインされている SWF コンテンツの中には、キーボードを使用してアプリケーションを制御するものがあります。ただし、iPhone で仮想キーボードが存在するのは、ユーザーがテキストフィールドを編集するときのみです。iPhone アプリケーションがキーボードイベントを送出するのは、仮想キーボードが存在するときのみです。

## アプリケーションの状態の保存

アプリケーションはいつ終了するかわかりません（電話がかかってきたときなど）。アプリケーションの状態が変更されるたびに新しい状態を保存することを検討してください。例えば、アプリケーション格納ディレクトリのファイルまたはデータベースに設定を保存することができます。または、ローカル共有オブジェクトにデータを保存することもできます。これにより、アプリケーションが再起動するときに状態を復元することができます。電話によってアプリケーションが中断された場合は、通話が終了すると再開します。

アプリケーション終了時に exiting イベントを送出する NativeApplication オブジェクトに依存しないでください。このイベントは送出不される場合があります。

## 画面の向きの変更

iPhone コンテンツは、縦長モードまたは横長モードで表示できます。アプリケーションが画面の向きの変更に応じてどのように対応するかを検査します。詳しくは、「[スクリーンの向きの変更の検出](#)」を参照してください。

## ヒットターゲット

ユーザーがタップするボタンや他のユーザーインターフェイスをデザインする際には、ヒットターゲットのサイズを検査します。タッチスクリーン上で指で簡単に起動できるように、十分な大きさにする必要があります。また、ターゲット間に十分なスペースも確保します。ヒットターゲットは 44 ~ 57 ピクセル程度である必要があります。

## メモリの割り当て

新しいメモリブロックの割り当てには負荷がかかります。アプリケーションの速度が低下する場合や、ガベージコレクションがトリガーされたときにアニメーションや操作の途中でパフォーマンスが低下する場合があります。

オブジェクトは、削除して新しいものを作成するのではなく、可能な限り再利用してください。

ベクトルオブジェクトは、配列よりもメモリの消費が少ないことを念頭においてください。「[Vector クラスと Array クラス](#)」を参照してください。

メモリの使用について詳しくは、「[メモリの節約](#)」を参照してください。

## 描画 API

ActionScript 描画 API (Graphics クラス) を使用してグラフィックを作成することは避けてください。描画 API を使用すると、ステージ上でオブジェクトが動的に作成され、ラスターライザーにレンダリングされます。可能であれば、これらのオブジェクトは、オーサリング時にステージで静的に作成します。

描画 API を使用して作成されたオブジェクトは、繰り返し呼び出されると、ActionScript が実行されるたびに破棄されて再作成されます。静的オブジェクトは、別のタイムラインに移動してもメモリ内に存在します。

## イベントバブリング

表示オブジェクトコンテナのネストが深い場合、イベントのバブリングで負荷がかかることがあります。この負荷を削減するには、イベントをターゲットオブジェクト内で完全に処理してから、イベントオブジェクトの `stopPropagation()` メソッドを呼び出します。このメソッドを呼び出すと、イベントのバブリングを回避できます。このメソッドを呼び出すことは、親オブジェクトがイベントを受け取らないことも意味します。

関連する効果は、表示オブジェクトのネストを浅くして、イベントチェーンを短縮することによって実現できます。

可能であれば、TouchEvent イベントの代わりに MouseEvent イベントを登録します。MouseEvent イベントでは TouchEvent イベントよりもプロセッサのオーバーヘッドが少なくなります。

可能であれば、`mouseEnabled` および `mouseChildren` プロパティを `false` に設定します。

## ビデオパフォーマンスの最適化

モバイルのビデオ再生を最適化するには、ビデオの再生中にアプリケーションが実行するその他の処理を最小限に抑えます。これにより、ビデオのデコード処理およびレンダリング処理で CPU を最大限に使用できます。

ビデオの再生中は、できる限り ActionScript コードを実行しないようにします。タイマー間隔が短いコード、またはタイムライン上で実行するコードは、実行しないようにします。

ビデオ以外の表示オブジェクトの再描画を最小限に抑えます。特に、ビデオ領域と交差するような表示オブジェクトの再描画は避けるようにしてください。これは、ビデオの下に隠れている表示オブジェクトの場合でもあてはまります。これらのオブジェクトは、引き続き再描画され、多くの処理リソースを消費します。例えば、位置インジケータには単純なシェイプを使用し、位置インジケータをフレームごとに更新するのではなく、1 秒間に 2～3 回だけ更新するようにします。ビデオコントロールはビデオ領域の直下に配置し、重なり合うことがないようにします。ビデオバッファリングアニメーションがある場合、使用していないときはビデオの背面に隠すのではなく、非表示に設定します。

## Flex と Flash のコンポーネント

多くの Flex コンポーネントおよび Flash コンポーネントはデスクトップアプリケーションで使用する目的で設計されています。これらのコンポーネント、特に表示コンポーネントはモバイルデバイス上での動作が遅くなる可能性があります。動作が遅くなるのに加えて、デスクトップコンポーネントがモバイルデバイス向けには不適切な操作モデルを含んでいる可能性があります。

Adobe は Flex フレームワークのモバイル最適化バージョンを開発しています。詳しくは、<http://labs.adobe.com/technologies/flex/mobile/> を参照してください。

## アプリケーションのファイルサイズの抑制

IPA ファイルのファイルサイズを縮小する方法を以下に示します。

- 背景ビットマップをチェックし、サイズが正しい（必要以上に大きくない）ことを確認します。
- 不要なフォントが埋め込まれていないかどうかを確認します。
- PNG アセットにアルファチャンネルがあるかどうかを調べ、不要な場合は削除します。PNG 圧縮などのユーティリティを使用して、PNG アセットのサイズを縮小します。
- PNG アセットを JPG アセットに変換します（可能な場合）。
- 音声ファイルの圧縮を検討します（低ビットレートを使用する）。
- 使用していないアセットを削除します。