
iAdプログラミングガイド





Apple Inc.
© 2011 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製複製（コピー）することを禁じます。また、製品に付属のソフトウェアは同梱のソフトウェア使用許諾契約書に記載の条件のもとでお使いください。書類を個人で使用する場合に限り1台のコンピュータに保管すること、またその書類にアップルの著作権表示が含まれる限り、個人的な利用を目的に書類を複製することを認めます。

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

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

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

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

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

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

iAd is a service mark of Apple Inc.

Apple, the Apple logo, iPhone, iTunes, Objective-C, and Xcode are trademarks of Apple Inc., registered in the United States and other countries.

iPad 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.

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。

目次

序章

iAdの概要 7

概要 7

アプリケーションでiAdを使用するには、iAd Networkに参加する必要があります。 7

Banner Viewは画面の一部を使用してバナー広告を表示する 8

フルスクリーン広告はiPadアプリケーションに対してさらに大きな広告を提供する 8

ユーザが広告とやり取りしている時は重要でないアクティビティを一時停止する 8

広告の取り消しはアプリケーションにはマイナス影響 8

アプリケーションをリリースする前にiAd Supportを検証する 9

お読みになる前に 9

関連項目 9

第1章

Banner Viewの概念 11

Banner ViewではView Controllerを必要とする 12

Banner Viewの作成 12

Banner Viewのサイズ 12

スレッド安全性 13

第2章

Banner Viewでの作業 15

バナーイベントへの応答 15

Banner Viewでのタッチへの応答 15

広告をロードしたときの応答 16

エラー処理 17

広告アクションの取り消し 18

バナーサイズの動的変更 18

Banner Viewベストプラクティス 19

第3章

フルスクリーン広告 21

フルスクリーン広告は、iPadでのみ利用できます。 21

フルスクリーン広告の概念 21

コンテンツとしてフルスクリーン広告を表示する 21

トランジション画面としてフルスクリーン広告を表示する 22

Banner Viewと異なるフルスクリーン広告のプログラムモデル 22

フルスクリーン広告オブジェクトのライフサイクル 23

広告オブジェクトの作成 24

広告の表示 25

ユーザによる広告とのやり取りの処理 26

広告オブジェクトのコンテンツは一定時間しか存続しません 28

フルスクリーン広告エラーの処理 28

第 4 章 **iAdアプリケーションのテスト 31**

- バナー広告のテスト 31
- チェックリストのテスト 31
 - Banner View 31
- フルスクリーン広告 32

改訂履歴 **書類の改訂履歴 33**

図、表、リスト

第 1 章 **Banner View の概念 11**

- 図 1-1 Banner View 上の広告 11
- 図 1-2 バナーアクション 11
- リスト 1-1 プログラムによる縦長表示の Banner View の作成 12
- リスト 1-2 プログラムによる横長表示の Banner View の作成 13

第 2 章 **Banner View での作業 15**

- リスト 2-1 アクションのトリガを許可する 16
- リスト 2-2 新しい広告がロードされた後の Banner View でのアニメーション 17
- リスト 2-3 広告が無効な場合の Banner View の削除 17
- リスト 2-4 Banner View で横向きおよび縦向きに対処するための設定 18
- リスト 2-5 向きの変化に応答する 18

第 3 章 **フルスクリーン広告 21**

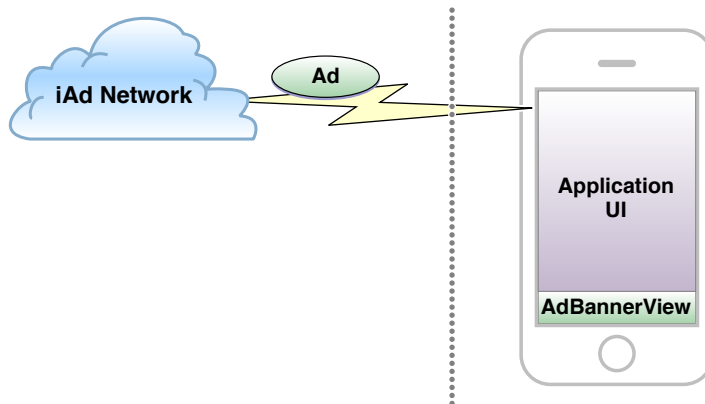
- 図 3-1 モードレスページでのフルスクリーン広告 22
- 図 3-2 モーダルページでのフルスクリーン広告 22
- 図 3-3 アプリケーションがフルスクリーン広告とやり取りする方法 23
- リスト 3-1 フルスクリーン広告オブジェクトの作成 24
- リスト 3-2 モーダルで広告を表示 25
- リスト 3-3 Scroll View への広告ページの追加 26
- リスト 3-4 アクションのトリガを許可する 27
- リスト 3-5 広告オブジェクトがコンテンツを解放した後にその広告オブジェクトを解放する 28
- リスト 3-6 フルスクリーン広告エラーの記録 29

第 4 章 **iAd アプリケーションのテスト 31**

- 表 4-1 アプリケーションで表示される広告 31

iAdの概要

iAd広告プラットフォームは、収益を得て、アプリケーションを促進する新たな機会をデベロッパにもたらしめます。アプリケーションのユーザインターフェイスにバナーやフルスクリーン広告を追加できます。Appleは広告領域を販売し、その領域に広告を配布します。作成したアプリケーションで表示した広告をユーザが見たり、やり取りしたときに収益を得ることができます。



概要

iAdフレームワークはiOS 4.0に導入され、iAd Networkから広告をダウンロードするために必要な動作を行います。主要な責務は、広告を入れるようにユーザインターフェイスを設計することです。

アプリケーションでiAdを使用するには、iAd Networkに参加する必要があります。

広告サポートをアプリケーションに追加する前に、iAd Network契約に同意する必要があります。さらに、iTunes Connectで各アプリケーションに対してiAdを明示的に有効にする必要があります。iAd Networkの一環として、アプリケーションに送付される広告の種類を制御できます。

iAd Networkの詳細については、<http://developer.apple.com/iad/>を参照してください。

アプリケーションでiAdを有効にする方法については、『iTunes Connect Developer Guide』を参照してください。

Banner Viewは画面の一部を使用してバナー広告を表示する

ADBannerViewクラスを使って、アプリケーションのユーザインターフェイスの一部をバナー広告の表示用に確保できます。BannerViewは、作成されると新しい広告を自動的にダウンロードし、ユーザに表示します。ユーザはバナーをタップして、広告のコンテンツとやり取りします。

関連する章： [「Banner Viewの概念」](#) (11 ページ)

フルスクリーン広告はiPadアプリケーションに対してさらに大きな広告を提供する

ADInterstitialAdクラスを使用すると、アプリケーションでフルスクリーン広告を表示できます。アプリケーションで提供したコンテンツのほかのページと一緒にコンテンツのページとしてフルスクリーン広告を追加できます。あるいは、アプリケーションがある画面から別の画面に移行するときに、フルスクリーン広告をモーダルモードで表示できます。バナー広告と同様に、ユーザはフルスクリーン広告をタップして、広告内容としてのリッチコンテンツを立ち上げます。

関連する章： [「フルスクリーン広告」](#) (21 ページ)

ユーザが広告とやり取りしている時は重要でないアクティビティを一時停止する

ユーザが広告をタップすると、iAdは一般的に画面を覆い、インタラクティブな広告を表示します。アプリケーションは実行し続けますが、ユーザは目で確認したり、アプリケーションのユーザインターフェイスとやり取りすることはできません。代わりに、広告の表示による対話型のリッチメディア体験をユーザに提供します。広告が表示されている間、アプリケーションのアクティビティを減らし、アプリケーションのユーザインターフェイスとのやり取りを必要とする機能はすべて一時停止する必要があります。アプリケーションはメモリ不足の警告に応答する必要があります。ユーザが広告とのやり取りを終えた後、簡単に再作成できるオブジェクトを解放する必要があります。

関連する章： [「Banner Viewでの作業」](#) (15 ページ) および [「フルスクリーン広告」](#) (21 ページ)

広告の取り消しはアプリケーションにはマイナス影響

ユーザが広告とやり取りしている間、アプリケーションはイベントを受信し続けます。ユーザが広告とのやり取りを終了すると制御はアプリケーションに戻ります。ただし、ユーザに即座に注目してもらいたいイベントをアプリケーションが受信した場合は、プログラムで広告を取り消して、ユーザインターフェイスを復元することができます。早急にユーザの注意を引く必要がある場合、アプリケーションで広告だけを取り消してください。広告を繁雑に取り消すと、アプリケーションから得られる収入が減少します。アプリケーションに提供される広告のインベントリにも影響を与える場合もあります。

関連する章： 「[Banner Viewでの作業](#)」（15 ページ） および 「[フルスクリーン広告](#)」（21 ページ）

アプリケーションをリリースする前にiAd Supportを検証する

Xcodeでアプリケーションをビルドし、実行する場合、iAdは自動的にアプリケーションにテスト広告を提供します。『*iOS Human Interface Guidelines*』と同様にこのプログラミングガイドにも記載されているガイドラインと推奨事項をアプリケーションが正しくサポートしていることを確認する必要があります。

関連する章： 「[iAdアプリケーションのテスト](#)」（31 ページ）

お読みになる前に

このガイドは、Objective-Cでのプログラミングを熟知しているデベロッパーを対象としています。iAdはビューとView Controllerを使用して、広告を表示します。ユーザはビュープログラミングに精通しており、View Controllerを使用してユーザインターフェイスを管理することが必要です。ビュープログラミングの詳細については、『*View Programming Guide for iOS*』を参照してください。View Controllerの詳細については、『*View Controller Programming Guide for iOS*』を参照してください。

ユーザが広告をタップした結果、iAdはアプリケーションをバックグラウンドに移行し、ほかのアプリケーションを起動する場合があります。iOS 4.0に導入されたマルチタスクの動作をよく理解している必要があります。詳細については、『*iOS App Programming Guide*』を参照してください。

関連項目

バナー広告の使用法を示したサンプルコードについては、『*iAdSuite*』を参照してください。

フルスクリーン広告の使用法を示したサンプルコードについては、『*iAdInterstitialSuite*』を参照してください。

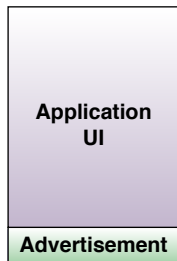
アプリケーションで広告を表示する方法のガイダンスや制約事項については、『*iOS Human Interface Guidelines*』を参照してください。

iAdフレームワークでのクラスやプロトコルの詳細については、『*iAdFrameworkReference*』を参照してください。

Banner Viewの概念

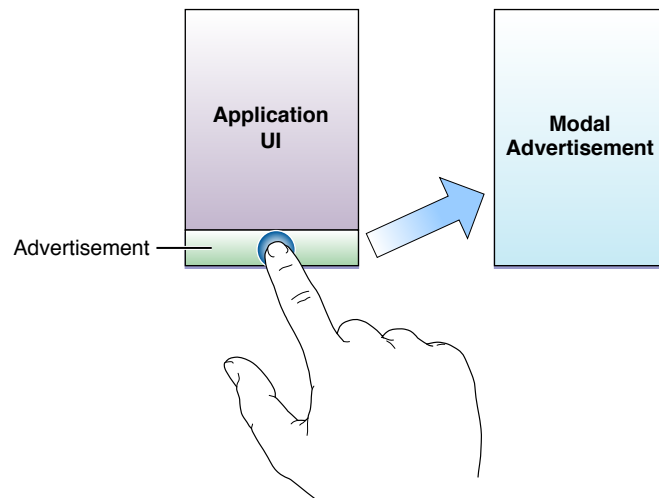
Banner Viewは定期的にiAd Networkから広告を取得し、ユーザに表示します。アプリケーションは、図 1-1に示すように、ユーザインターフェイス画面の一部をBanner View用に確保します。

図 1-1 Banner View上の広告



広告ではユーザがバナーをタップした際に起こすアクションを定義します。たとえば、別のアプリケーションを起動したり、アプリケーションのユーザインターフェイスを一時的に覆って対話的広告を表示したりすることが考えられます。図 1-2にユーザがディスプレイをタップした際の画面の遷移を示します。

図 1-2 バナーアクション



ユーザは常にプロセスを制御しています。バナー広告に関連付けられたコンテンツをいつ表示するかを決定するのはユーザだけです。

Banner View では View Controller を必要とする

Banner View を含むユーザインターフェイス画面はすべて **View Controller** (UIViewController をサブクラス化するクラス) で管理します。このクラスが、追加の広告画面でアプリケーションのユーザインターフェイスを覆うアクションを引き起こします。表示されているときの **Banner View** は常に、**View Controller** の `view` プロパティにアタッチされたビュー階層の一部でなければなりません。これを達成する最も簡単な方法は、**View Controller** のインターフェイスのインスタンス化に使用する nib ファイルの一部としてビューをインスタンス化することです。

Banner View の作成

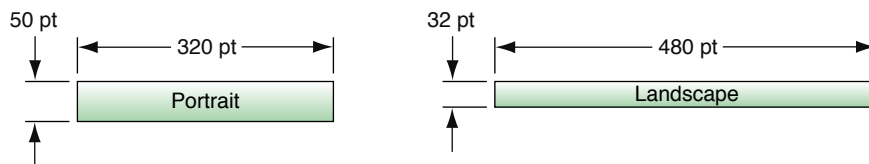
リスト 1-1 に、**View Controller** を使った、プログラミングによる縦長モードの **Banner View** を作成する最も単純なコードを示します。

リスト 1-1 プログラムによる縦長表示の **Banner View** の作成

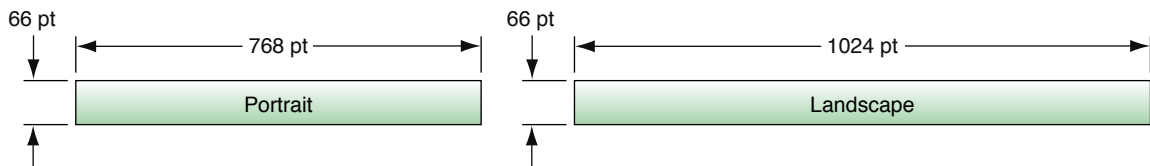
```
ADBannerView *adView = [[ADBannerView alloc] initWithFrame:CGRectZero];
adView.currentContentSizeIdentifier = ADBannerContentSizeIdentifierPortrait;
[self.view addSubview:adView];
```

Banner View のサイズ

iAd は、縦長と横長のアプリケーションに対してさまざまなバナーサイズをサポートしています。広告の正確なサイズは、バナーを表示するデバイスによって異なります。**iPhone** の場合、縦長の広告は 320 x 50 ポイント、横長の広告は 480 x 32 ポイントです。**iPad** の場合、縦長の広告は 768 x 66 ポイント、横長の広告は 1024 x 66 ポイントです。将来は、iAd で追加のサイズが公開される可能性があります。



iPhone iAd Sizes



iPad iAd Sizes

広告が確実に正しく表示されるように、固定の広告サイズの1つに、**Banner View**の大きさを常に一致させる必要があります。そのため、`ADBannerView`クラスでは、ユーザがフレームを直接変更できないようにしています。その代わりに`currentContentSizeIdentifier`プロパティを設定することで、**Banner View**のフレームを変更できます。このプロパティに設定されている値を変更すると、与えられた識別子のサイズに合うように**Banner View**のフレームサイズが変更されます。ユーザが特定のサイズ識別子を設定する前に、そのサイズ識別子が`requiredContentSizeIdentifiers`プロパティに設定されているサイズ識別子のセットに含まれている必要があります。たとえば、リスト 1-2 では、プログラムによる**View Controller**での横長の**Banner View**の作成方法を示しています。

リスト 1-2 プログラムによる横長表示のBanner Viewの作成

```
ADBannerView *adView = [[ADBannerView alloc] initWithFrame:CGRectZero];
adView.requiredContentSizeIdentifiers = [NSSet
setWithObject:ADBannerContentSizeIdentifierLandscape];
adView.currentContentSizeIdentifier = ADBannerContentSizeIdentifierLandscape;
[self.view addSubview:adView];
```

実行時に広告が使用する正確なサイズがアプリケーションで必要な場合は、`ADBannerContentSizeIdentifierLandscape`または`ADBannerContentSizeIdentifierPortrait`のいずれかを渡して、`sizeFromBannerContentSizeIdentifier:`クラスメソッドを呼び出します。

スレッド安全性

Banner Viewはユーザインターフェイス要素の1つです。その他のユーザインターフェイス要素と同様、**Banner View**は、アプリケーションのメインスレッドからのみ参照すべきです。

Banner Viewでの作業

Banner Viewはデリゲートを使用してアプリケーションと通信します。アプリケーションではBanner Viewデリゲートを実装してBanner Viewのライフサイクルでの共通のイベントを処理します。アプリケーションでは次が必要です。

- ユーザがバナーをタップしたときの応答
- Banner Viewが広告をロードしたときの応答
- Banner Viewがエラーに遭遇したときの応答

典型的なパターンでは、カスタムView Controllerでこれらのメソッドを実装しますが、ほかのオブジェクトで実装することもできます。

アプリケーションで向きの変化をサポートしている場合は、デバイスの向きが変わったらView ControllerでBanner Viewのサイズも変更しなければなりません。

バナーイベントへの応答

多くのBanner Viewデリゲートは次の動作のすべてを実装します。

Banner Viewでのタッチへの応答

広告アクションをトリガする前に、Banner Viewはデリゲートの `bannerViewActionShouldBegin:willLeaveApplication:` メソッドを呼び出します。デリゲートメソッドは、次の2つのタスクを実行します。

- トリガされたアクションを実行するか判断する
- アクションがアプリケーションのユーザインターフェイスを覆ってしまう場合、ユーザとのやり取りを必要とするアクティビティをこのメソッドが一時停止する

アクションがトリガされるのを許可する場合、デリゲートではこのメソッドからYESを返す必要があります。NOを返すとアクションをトリガさせないようにできます。この操作を安全に実行させることができない場合は、アプリケーションではトリガされたアクションを常に許可すべきです。

- `willLeave` パラメータがYESであれば、このデリゲートメソッドから復帰した後に、アプリケーションはバックグラウンドに移されます。このプロセスについては、「アプリケーションの状態および遷移の理解」 in *iOS App Programming Guide* で説明します。

- willLeaveパラメータがNOであれば、デリゲートメソッドから復帰した後に、iAdはアプリケーションのユーザインターフェイスを覆います。アプリケーションはメソッドから戻る前に、サウンド、アニメーション、またはユーザとのやり取りを必要とするその他のアクティビティを無効にする必要があります。たとえば、リアルタイムのゲームでは、ゲームプレイを一時停止した後、YESを返してトリガされたアクションを許可する必要があります。

「BannerViewの作成」は、アプリケーションがこのデリゲートメソッドを実装する方法の典型的なパターンを示します。

リスト 2-1 アクションのトリガを許可する

```
- (BOOL)bannerViewActionShouldBegin:(ADBannerView *)banner
willLeaveApplication:(BOOL)willLeave
{
    NSLog(@"Banner view is beginning an ad action");
    BOOL shouldExecuteAction = [self allowActionToRun]; // アプリケーションはこの方
    法を実装する
    if (!willLeave && shouldExecuteAction)
    {
        // ここにコードを挿入して、広告と競合する可能性のあるサービスを一時停止する
    }
    return shouldExecuteAction;
}
```

BannerViewがアプリケーションのユーザインターフェイスを覆っている場合は、BannerViewはインターフェイスが復元したらデリゲートのbannerViewActionDidFinish:メソッドを呼び出します。このメソッドを実装すると、アプリケーションで一時停止したサービスがすべて復元するはずで

重要： willLeaveパラメータがYESであったためにアプリケーションをバックグラウンドに移動した場合、アプリケーションのユーザインターフェイスはBanner Viewで覆われておらず、bannerViewActionDidFinish:呼び出しを受け取りません。一方、インターフェイスがBanner Viewで覆われていた場合、広告がほかのアプリケーションを起動する、またはユーザがそのような操作を選択するなどの理由で、アプリケーションが後でバックグラウンドに移動される可能性があります。こうしたすべての場合において、アプリケーションのユーザインターフェイスがBanner Viewで覆われていた場合は、覆われていたBanner Viewが除かれて、デリゲートのbannerViewActionDidFinish:が呼び出されてからアプリケーションはバックグラウンドに移されます。アプリケーションがバックグラウンドに移る可能性があるため、デリゲートはアプリケーションのbannerViewActionDidFinish:メソッドから素早く戻る必要があります。

4.2以前のiOSでは、ユーザが広告とやり取りをする間、アプリケーションはBanner Viewオブジェクトを削除すべきではありません。Banner ViewデリゲートのbannerViewActionDidFinish:メソッドが呼び出された後だけに、Banner Viewを削除します。

広告をロードしたときの応答

iAdフレームワークを使うと、非同期のモデルを採用して、利用可能な広告がある場合のみ表示を行うことが簡単にできます。アプリケーションは空のBannerViewを表示してはいけません。その代わりに、広告がある場合はバナーを表示する必要がありますが、表示するものがなければバナーを非表示にします。

新しく表示する広告がある場合、**Banner View**はデリゲートの**bannerViewDidLoadAd:**メソッドを呼び出します。このメソッドは、**Banner View**が現在ビュー階層の一部でなくても呼び出されます。アプリケーションはこのメソッドを使ってビューをビュー階層に追加するか、画面に**Banner View**を移動させることができます。「**Banner Viewのサイズ**」はプロパティを使用して、**Banner View**が表示されているかどうかを確認します。バナーが非表示で新しい広告がロードされている場合、メソッドはビューを画面にロードします。

リスト 2-2 新しい広告がロードされた後のBanner Viewでのアニメーション

```
- (void)bannerViewDidLoadAd:(ADBannerView *)banner
{
    if (!self.bannerIsVisible)
    {
        [UIView beginAnimations:@"animateAdBannerOn" context:NULL];
// Banner Viewが画面下部の直上にあると仮定する
        banner.frame = CGRectOffset(banner.frame, 0, -banner.frame.size.height);
        [UIView commitAnimations];
        self.bannerIsVisible = YES;
    }
}
```

エラー処理

エラーが発生したら、**Banner View**はデリゲートの**bannerView:didFailToReceiveAdWithError:**メソッドを呼び出します。これが発生したら、アプリケーションは**Banner View**を非表示にします。リスト 2-3は、実装方法の1つを示します。それは、[リスト 2-2](#) (17 ページ) と同じプロパティを使用して、バナーが表示されているかどうかを追跡します。バナーが表示されて、エラーが発生した場合、バナーを画面外まで移動します。

リスト 2-3 広告が無効な場合のBanner Viewの削除

```
- (void)bannerView:(ADBannerView *)banner didFailToReceiveAdWithError:(NSError *)error
{
    if (self.bannerIsVisible)
    {
        [UIView beginAnimations:@"animateAdBannerOff" context:NULL];
// Banner Viewが画面下部にあると仮定する
        banner.frame = CGRectOffset(banner.frame, 0, banner.frame.size.height);
        [UIView commitAnimations];
        self.bannerIsVisible = NO;
    }
}
```

エラーがデリゲートに送信された後でも、**Banner View**は新しい広告をダウンロードしようとし続けます。したがって、これらのデリゲートメソッドの両方を実装することで、アプリケーションは広告がロードされた場合だけにバナーを表示できます。

広告アクションの取り消し

アクションを実行するためにBanner Viewがアプリケーションのユーザインターフェイスを覆っているときも、アプリケーションはイベントを受信し続けます。アプリケーションは、Banner Viewの `bannerViewActionInProgress` プロパティを読み込んで、バナーのアクションが実行中かどうか判断できます。アプリケーションは自身のアクティビティを縮小し、ユーザとのやり取りを必要とするアクションを回避する必要があります。

ユーザの注意を引く必要があるイベントが発生した場合は、アプリケーションからBanner Viewの `cancelBannerViewAction` メソッドを呼び出して広告のアクションを取り消すことができます。アクションが終了したら、Banner Viewはデリゲートの `bannerViewActionDidFinish:` メソッドを呼び出します。

重要： 広告アクションを取り消したり実行しなかったりすると、アプリケーションが受け取る広告や、広告を表示することでデベロッパが受け取る収入に潜在的に影響する可能性があります。アプリケーションでアクションを取り消すのは、早急にユーザの注意を引く必要がある場合のみにすべきです。たとえば、VoIP (Voice over Internet Protocol)を提供するアプリケーションでは、アプリケーションでほかのユーザからの電話を受信した場合に広告を取り消すことがあります。

バナーサイズの動的変更

作成後にBanner Viewのサイズを変更する予定のアプリケーションは、アプリケーション内で設定する可能性のあるサイズをすべて、`requiredContentSizeIdentifiers` プロパティに一式設定する必要があります。アプリケーションで複数サイズをサポートする最も一般的な理由は、向きの変化をサポートするためです。向きの変化に対応してアプリケーションのインターフェイスを変更する場合、新しい向きに合わせてBanner Viewをリサイズする必要があります。

リスト2-4に、縦向きおよび横向きの両方の画像を持った広告をダウンロードするためのView Controllerの設定方法を示します。

リスト 2-4 Banner Viewで横向きおよび縦向きに対処するための設定

```
self.bannerView.requiredContentSizeIdentifiers = [NSSet setWithObjects:
    ADBannerContentSizeIdentifierPortrait, ADBannerContentSizeIdentifierLandscape,
    nil];
```

向きが変化すると、View Controllerの `willRotateToInterfaceOrientation:duration:` メソッドがバナーサイズを変更します。

リスト 2-5 向きの変化に応答する

```
-
(void)willRotateToInterfaceOrientation:(UIInterfaceOrientation)toInterfaceOrientation
duration:(NSTimeInterval)duration
{
    if (UIInterfaceOrientationIsLandscape(toInterfaceOrientation))
        self.bannerView.currentContentSizeIdentifier =
            ADBannerContentSizeIdentifierLandscape;
    else
        self.bannerView.currentContentSizeIdentifier =
```

```

        ADBannerContentSizeIdentifierPortrait;
    }

```

アプリケーションでrequiredContentSizeIdentifiersプロパティに1つ以上のバナーサイズを設定すると、iAdサービスでは指定されたすべてのサイズの画像を提供する広告のみダウンロードします。これにより、アプリケーションがビューのサイズを変更したときに、Banner Viewでは表示した広告をシームレスに変更できるようになります。このことでBanner Viewで利用できる広告が限られるため、デベロッパーがrequiredContentSizeIdentifiersプロパティに設定するサイズはアプリケーションで実際に利用するものだけにする必要があります。

注： 現在のバージョンのiAdは縦向きおよび横向きの広告のみサポートします。そのため、この制限は現時点ではあまり意味がありません。実際には、Appleはほとんどの広告主が縦向きと横向きの両方のビューを提供することを想定しています。そのため、requiredContentSizeIdentifiersプロパティに両方を含めても、広告のインベントリが大きく制限されることはありません。将来、ほかのサイズが追加された場合は、使用されていないコンテンツサイズを指定すると、アプリケーションで利用できる広告をより狭い範囲に不要に制限する可能性があります。

Banner Viewベストプラクティス

アプリケーションを設計する場合は、次の原則を念頭に置いてください。

- Banner Viewを作成するのは、実際に表示してユーザに見せる場合だけにしてください。そうしないと、広告が繰り返し表示され、アプリケーションに表示できる広告のリストを使い果たしてしまうおそれがあります。
- ユーザが、Banner Viewが付随する画面からそうでない画面にナビゲートする場合、このあと長時間にわたってその画面にとどまると予想されるならば、新しい画面に遷移する前に、ビュー階層からBanner Viewを削除し、そのデリゲートとしてnilを設定して、解放してください。要するに、ユーザの目に触れないならば、Banner Viewをいつまでも保持しておかない、ということです。
- アプリケーションがBanner Viewを作成する場合、ビューが実際に広告を表示するまでに遅延が生じます。ユーザに対して短い時間しか表示されないコンテンツの画面のBanner Viewを使用する場合、ユーザが画面のコンテンツとのやり取りを終えるまでの間では、バナーが広告をダウンロードする時間は十分ではない可能性があります。その代わりに、アプリケーションは1つのBanner Viewを作成し、ユーザインターフェイス全体で使用します。ユーザがアプリケーションを移動すると、アプリケーションはバナーの表示が期待される任意の画面にBanner Viewを移動させます。『iAdSuite』サンプルはこの技術を実装する方法を示します。
- 広告がリッチメディア体験に移行する場合、ユーザに対話型広告を表示するために、iAdはさらに多くのメモリを消費します。このメモリは、アプリケーションで利用可能なメモリから割り当てられます。アプリケーションは、そのアクティビティを縮小して、広告がスムーズに動作できるようにします。ユーザが広告とのやり取りを終了した後に簡単に再構築できる大きなオブジェクトを解放することで、メモリ不足状態に迅速に対応できるようにします。

フルスクリーン広告

フルスクリーン広告は、iOS 4.3に導入された新しい種類の広告です。フルスクリーン広告は、`ADInterstitialAd`クラスによって実装されています。

フルスクリーン広告は、iPadでのみ利用できます。

フルスクリーン広告が利用できるかどうかを確認するには、`UI_USER_INTERFACE_IDIOM`マクロの値を取得します。このマクロから返された値が`UIUserInterfaceIdiomPad`ならば、フルスクリーン広告を使用できます。

フルスクリーン広告の概念

iAdが提供するフルスクリーン広告には、2つの一般的な用途があります。

- 広告は、アプリケーションが提供するコンテンツと同様にコンテンツのページとして表示されます。通常、これは`ScrollView`のコンテンツにフルスクリーン広告を追加することを意味します。
- アプリケーションの2箇所間のトランジション画面として

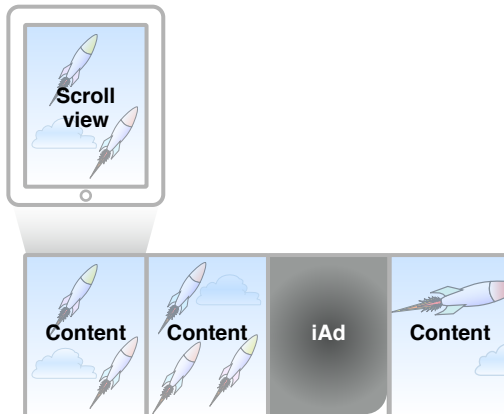
アプリケーションへのフルスクリーン広告の追加に着手する前に、アプリケーションに導入しようとしている各使用事例に対して、アプリケーションがどのように動作するかを理解している必要があります。『*iAdInterstitialSuite*』サンプルには、両方の使用事例が含まれます。

コンテンツとしてフルスクリーン広告を表示する

この使用事例では、広告はアプリケーションが提供するほかのコンテンツと同列に表示されます。広告の内容はスクリーンサイズのページに組み込まれ、コンテンツ表示領域をページ境界までスクロールするように構成されている`ScrollView`に配置されます。[図 3-1](#)（22 ページ）に、この概念を説明しています。`ScrollView`には4つのビューがあります。3つのビューはアプリケーションのコンテンツを保持していますが、4つ目のビューはフルスクリーン広告を保持しています。ユーザは左または右にスワイプして、コンテンツを移動できます。ユーザがiAdをタップしたら、広告によるリッチメディア体験が開始されます。

アプリケーションが自身のコンテンツと同列に並べてフルスクリーン広告を表示した場合、ユーザが広告とのやり取りを選択したときのみ、iAdはアプリケーション内のほかの動作をブロックしません。

図 3-1 モードレスページでのフルスクリーン広告

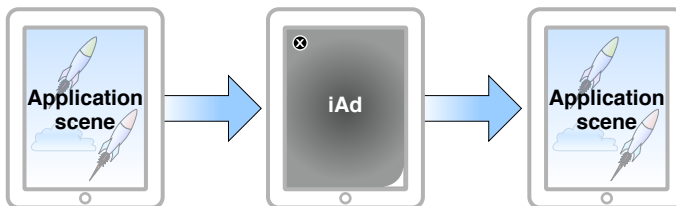


トランジション画面としてフルスクリーン広告を表示する

この使用事例では、広告はアプリケーションで表示した2つの別の画面の間にトランジション画面として表示されます。図 3-2（22 ページ）に、この動作の例を示します。たとえば、

『iAdInterstitialSuite』のゲームサンプルは、プレイヤーが対戦を終了してからほかの対戦を始める前に広告を表示します。広告がトランジションページとして表示されたら、広告はモーダルで表示されます。ユーザは広告とやり取りします。アプリケーションで次の画面に移動するには、ユーザは広告の閉じるボタンを明示的にタップします。

図 3-2 モーダルページでのフルスクリーン広告



Banner Viewと異なるフルスクリーン広告のプログラムモデル

ADInterstitialAdオブジェクト向けのプログラムモデルは、ADBannerViewオブジェクト向けのプログラムモデルと同様ですが、2点重要な違いがあります。

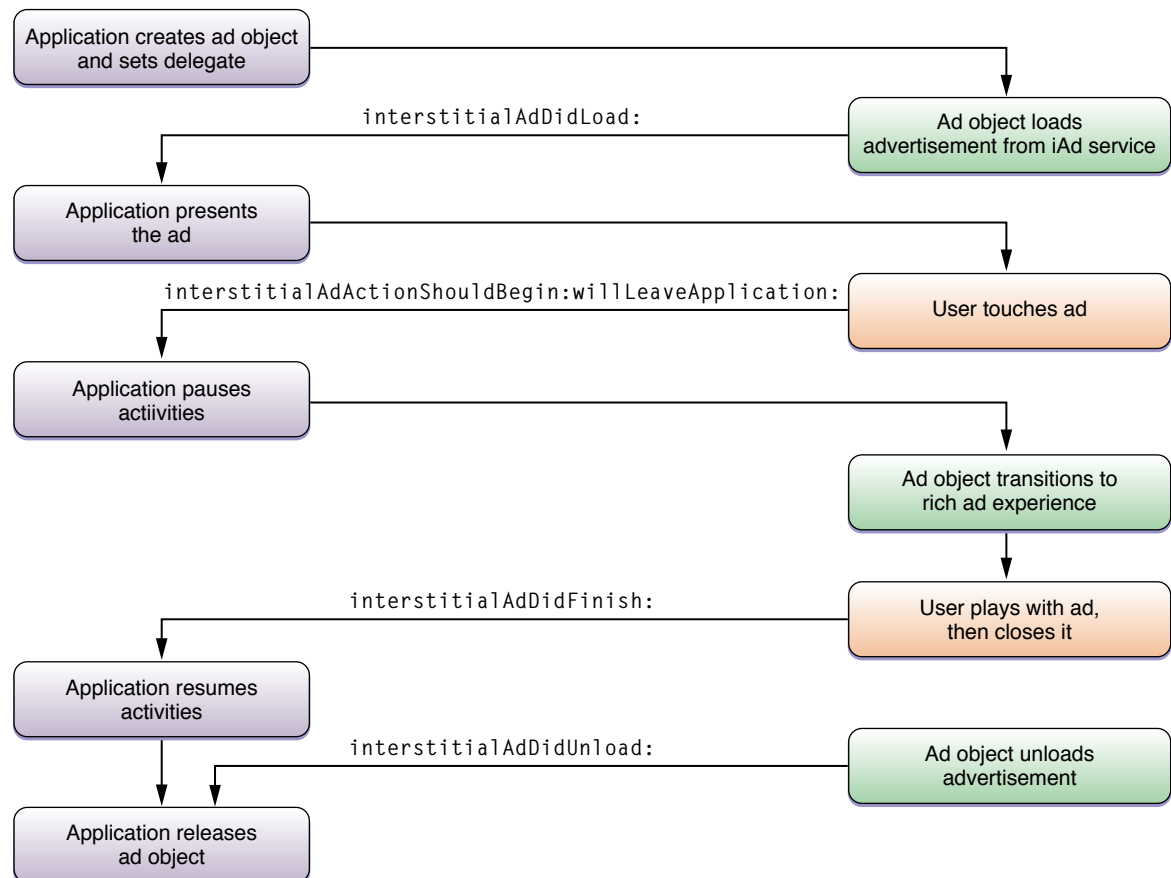
ADInterstitialAdオブジェクトは実際のビューではありません。広告オブジェクトはそのコンテンツのダウンロードを管理し、広告を表示するビューを作成できます。しかし、フルスクリーン画面広告を表示するには、アプリケーションは明示的に広告を表示します。アプリケーションがモーダル広告を表示する場合、View Controllerを使用して広告を表示します。それに対して、モードレスページとして広告を追加する場合、アプリケーションは適切な大きさとカスタムビューを作成し、View Controllerに関連したビュー階層に追加します。そして、アプリケーションはこのビューを使用して広告を表示します。iAdはこのビューを使用して、広告のコンテンツを保持します。

バナービューとフルスクリーン広告の2つ目の主要な違いは、フルスクリーン広告は、新しいコンテンツを通したサイクルを行わない点です。フルスクリーン広告オブジェクトは、1つの広告をロードします。1つのコンテンツの有効期限が切れたら、アプリケーションは広告オブジェクトを解放しなければなりません。新しい広告を表示する必要があるたびに、アプリケーションは明示的に新しい広告オブジェクトを作成しなければなりません。

フルスクリーン広告オブジェクトのライフサイクル

図 3-3 に、アプリケーション内にフルスクリーン広告を実装するプロセスを示しています。ライフサイクルは、アプリケーションが `ADInterstitialAd` オブジェクトを作成し、デリゲートに設定したときに始まります。広告オブジェクトはライフサイクルの重要な点でデリゲートにメッセージを送信します。そのためデリゲートは必要です。図 3-3 に、基本的で重要なデリゲートメソッドの多くを示します。

図 3-3 アプリケーションがフルスクリーン広告とやり取りする方法



広告オブジェクトが作成されたら、`iAdNetwork` から広告を自動的にダウンロードします。広告のダウンロードが終了したら、オブジェクトはデリゲートにシグナルを送ります。アプリケーションはロードした広告を取得し、ユーザに表示します。アプリケーションが使用する実際のプロセスが広告を表示する方法は、モーダルページに表示するか、モードレスページに表示するかによって異なります。

広告にタッチすると、アプリケーションに通知されるので、広告に影響を与える可能性があるすべてのアクティビティを一時停止できます。この手順は、**BannerView**がアプリケーションにユーザの興味を通知する手順と同様です。アプリケーションがそのアクティビティをいったん停止した後、広告オブジェクトは*iAd Network*からインタラクティブな広告をロードし、ユーザに表示します。ユーザが広告とのやり取りを終了したら、広告オブジェクトはデリゲートに通知するので、アプリケーションは一時停止していたアクティビティを再開できます。

広告オブジェクトを作成した後のある時点で、広告のコンテンツの期限が切れます。コンテンツの期限は、広告がどのように表示されたか、ユーザがアプリケーションとどのようにやり取りしたかによって異なります。広告オブジェクトのコンテンツが解放されたら、デリゲートが呼び出されます。アプリケーションは広告オブジェクトを解放する必要がありますが、その時点でほかのタスクを実行していることがあります。たとえば、『*iAdInterstitialSuite*』にあるADGameの例は、新しい広告オブジェクトを作成してサイクルを再度繰り返します。それに対して、ADMagazineの例では、広告のビューがなくなってできた空白を埋めるために、これを削除してほかのコンテンツページに移動します。

次に、上記のダイアグラムの各ステップを実装する方法について説明します。

広告オブジェクトの作成

リスト 3-1 (24 ページ) に、フルスクリーン広告オブジェクトの作成を実装する一般的な方法を示します。広告オブジェクトには、ほかのプロパティを設定する必要はありません。

リスト 3-1 フルスクリーン広告オブジェクトの作成

```
interstitial = [[ADInterstitialAd alloc] init];
interstitial.delegate = self;
```

広告オブジェクトが作成されたら、*iAd Network*からの広告のダウンロードが自動的に開始されます。アプリケーションは次のどちらかの方法で、広告オブジェクトがそのコンテンツを正常にロードしたかどうかを判断します。

- アプリケーションは、オブジェクトがロードされた広告を保持しているかどうかを記載している、広告オブジェクトのloadedプロパティを読み込むことができます。広告を表示できるかどうかを判断するために、この値を定期的に呼び出さないでください。しかし、アプリケーションがコンテンツの新しい画面に移行するときにこのプロパティの値をチェックするのは有益です。特にいつ広告を表示するかがオプションである場合には便利です。たとえば、『*iAdInterstitialSuite*』のADGameの例は、対戦が終了した後の広告オブジェクトのloadedプロパティをテストしています。このプロパティの値がYESである場合、広告を表示します。値がNOである場合、ゲームは広告をスキップして、新しい対戦を開始します。
- デリゲートは広告オブジェクトが広告コンテンツをロードしたらすぐに通知するようにinterstitialAdDidLoad:メソッドを実装できます。一般的に、デリゲートメソッドは広告を表示します。デリゲートメソッドの実装は、広告がロードされたらすぐにアプリケーションのコンテンツに挿入したい場合に最適です。このメソッドを使用してすぐにユーザにモーダル広告を表示してはいけません。広告コンテンツがロードされたときにユーザが実行中のタスクが中断される可能性があるからです。

重要： 広告のコンテンツの有効期限は広告オブジェクトがその内容のロードを完了したときに基
づいて決まるので、同時に複数の広告オブジェクトを作成しないでください。複数の広告オブジェ
クトを同時に作成（広告の準備プールを作成）する場合、いくつかの広告は表示する前に期限切れ
になる可能性があります。

広告の表示

広告がそのコンテンツをロードしたら、ユーザに広告を表示できます。使用するプロセスは、コン
テンツをモーダルページで表示するか、モードレスページで表示するかによって決まります。

モーダルで広告を表示

広告は、**View Controller**オブジェクトによってモーダルで表示されます。アプリケーションは**View
Controller**を作成して、**View Controller**を唯一のパラメータとして渡し、広告オブジェクトの
`presentFromViewController:`メソッドを呼び出します。[リスト 3-2](#)（25 ページ）に、この動作の
最小限の実装を示します。実際では、広告がモーダルでアプリケーションの最上部に表示されるた
め、広告が表示される前に、アプリケーションでほかのアクティビティを一時停止することになり
ます。

リスト 3-2 モーダルで広告を表示

```
if (interstitial.loaded)
{
    [interstitial presentFromViewController:self];
}
```

広告を表示した後は、アプリケーションの**View Controller**は最前面の**View Controller**でなくなります。
広告オブジェクトは**View Controller**を作成して、広告の表示に使用します。ユーザが広告を消した
後、広告の**View Controller**は削除されます。

ページのコンテンツとして広告を表示

アプリケーションは、広告の親となるビューを作成することで広告をモードレスに表示します。こ
のビューオブジェクトは次の規則に従う必要があります。

- ビューは、**View Controller**オブジェクトのビュー階層に含まれる必要があります。この制限は、**Banner View**で必須の動作と同じです。
- ビューの幅は画面の幅と等しくなければなりません。
- ビューの高さは、画面の高さから113ポイント引いた高さ以上で、画面の高さより小さくなら
なければなりません。たとえば、iPadが横長モードの場合は、画面の高さは768ポイントです。した
がって、親となるビューの高さは、655ポイントと768ポイントの間になければなりません。
- **ViewController**が向きの変更をサポートしている場合、ビューは新しい方向に合わせるためサイ
ズを変更する必要があります。
- ビューはサブビューの親となることができません。

リスト 3-3に、一般的な実装を示します。新しいページへのインデックスを使用して、ページのフレームの計算、フレームを使用したビューのアロケート、およびScroll Viewへの追加を行います。そして、広告のpresentInView:メソッドを呼び出して、広告を新しいビューに関連付けます。

リスト 3-3 Scroll Viewへの広告ページの追加

```
CGRect interstitialFrame = scrollView.bounds;

interstitialFrame.origin = CGPointMake(interstitialFrame.size.width * index,
0);

UIView *view = [[UIView alloc] initWithFrame:interstitialFrame];

[scrollView addSubview:view];
[interstitial presentInView:view];
```

重要： ビューが広告の親となる場合、広告オブジェクトはビューにサブビューを追加します。このビューを変更しようとしたり、ほかのビューを追加して隠してはいけません。広告オブジェクトのコンテンツがビューによって表示される間、ビューのコンテンツをほかの広告オブジェクトが所有しているように取り扱う必要があります。ビューの位置をその親ビュー内で変更できます。これによって、ビューをScroll Viewの別のページに移動することができます。

ユーザによる広告とのやり取りの処理

ユーザが広告をタップすると、アクションが起こります。この動作は、Banner Viewの動作と同じです。

広告アクションの開始

アクションをトリガする前に、Banner ViewはデリゲートのinterstitialAdActionShouldBegin:willLeaveApplication:メソッドを呼び出します。デリゲートメソッドは、次の2つのタスクを実行します。

- トリガされたアクションを実行するか判断する
- アクションがアプリケーションのユーザインターフェイスを覆ってしまう場合、ユーザとのやり取りを必要とするアクティビティをこのメソッドが一時停止する

アクションがトリガされるのを許可する場合、デリゲートではこのメソッドからYESを返す必要があります。NOを返すとアクションをトリガさせないようにできます。この操作を安全に実行させることができない場合は、アプリケーションではトリガされたアクションを常に許可すべきです。

- willLeaveパラメータがYESであれば、このデリゲートメソッドから戻った後、アプリケーションはバックグラウンドに移されます。このプロセスについては、「アプリケーションの状態および遷移の理解」 in *iOS App Programming Guide*で説明します。
- willLeaveパラメータがNOであれば、デリゲートメソッドから復帰した後に、iAdはアプリケーションのユーザインターフェイスを覆います。アプリケーションでは、サウンド、アニメーション、またはユーザとのやり取りを必要とするその他のアクティビティを無効にする必要があります。たとえば、リアルタイムのゲームでは、トリガされたアクションを許可する前にゲームプレイを一時停止する必要があります。

リスト 3-4に、アプリケーションでこのデリゲートメソッドを実装する方法の構成について示します。

リスト 3-4 アクションのトリガを許可する

```
- (BOOL)interstitialAdActionShouldBegin:(ADInterstitialAd *)banner
willLeaveApplication:(BOOL)willLeave
{
    NSLog(@"Interstitial ad is beginning an ad action");
    BOOL shouldExecuteAction = [self allowActionToRun]; // アプリケーションはこの方
    法を実装する
    if (!willLeave && shouldExecuteAction)
    {
        // ここにコードを挿入して、広告と競合する可能性のあるサービスを一時停止する
    }
    return shouldExecuteAction;
}
```

広告アクションの完了

フルスクリーン広告がリッチメディア広告をアプリケーション内で表示した場合、広告が終了した後、デリゲートの`interstitialAdActionDidFinish:`メソッドを呼び出します。このメソッドを実装して、アクションが開始したときにアプリケーションで一時停止したサービスをすべて復元させます。

重要： `willLeave`パラメータがYESであったためにアプリケーションをバックグラウンドに移動した場合、アプリケーションのユーザインターフェイスはBanner Viewで覆われておらず、`interstitialAdActionDidFinish:`呼び出しを受け取りません。一方、インターフェイスがBanner Viewで覆われていた場合、広告がほかのアプリケーションを起動する、またはユーザがそのような操作を選択するなどの理由で、アプリケーションが後でバックグラウンドに移動される可能性があります。こうしたすべての場合において、アプリケーションのユーザインターフェイスがBanner Viewで覆われていた場合は、覆われていたBanner Viewが除かれて、デリゲートの`interstitialAdActionDidFinish:`が呼び出されてからアプリケーションはバックグラウンドに移されます。アプリケーションがバックグラウンドに移る可能性があるため、デリゲートはアプリケーションの`interstitialAdActionDidFinish:`メソッドから素早く戻る必要があります。

広告アクションの取り消し

フルスクリーン広告がそのアクションを実行したときに、アプリケーションは継続してイベントを受け取ります。アプリケーションは、広告オブジェクトの`actionInProgress`プロパティを読み込んで、アクションが実行中かどうかを判断できます。広告が実行されている間、アプリケーションは自身のアクティビティを縮小し、ユーザとのやり取りを必要とするアクションを回避する必要があります。

ユーザの注意を求めるイベントが発生した場合は、アプリケーションから広告オブジェクトの`cancelAction`メソッドを呼び出して広告のアクションを取り消すことができます。アクションはすぐに終了します。デリゲートの`interstitialAdActionDidFinish:`が呼び出されませんが、アプリケーションは、広告アクションが開始される前に一時停止したサービスを再開します。

重要： 広告アクションを取り消したり実行しなかったりすると、アプリケーションが受け取る広告や、広告を表示することでデベロッパが受け取る収入に潜在的に影響する可能性があります。アプリケーションでアクションを取り消すのは、早急にユーザの注意を引く必要がある場合のみにすべきです。たとえば、VoIP (Voice over Internet Protocol)を提供するアプリケーションでは、アプリケーションでほかのユーザからの電話を受信した場合に広告を取り消すことがあります。

広告オブジェクトのコンテンツは一定時間しか存続しません

広告オブジェクトによって提供されたコンテンツはいずれ終了します。終了すると、オブジェクトはコンテンツを解放し、デリゲートを呼び出します。広告のコンテンツの有効期限がいつ終了するかは、アプリケーションで制御できません。コンテンツの有効期限が切れる一般的なケースには次のようなものがあります。

- 広告オブジェクトにエラーが発生した場合、広告オブジェクトはエラーをアプリケーションに報告し、そのコンテンツを解放します。
- 広告オブジェクトがビューまたはView Controllerで表示されなかった場合、そのコンテンツは5分から15分後に無効になります。
- 広告オブジェクトがモードレスダイアログで表示された場合、そのコンテンツも5分から15分間有効です。広告オブジェクトに関連するビューがオフスクリーンの場合は、広告オブジェクトはコンテンツの期限が切れたときに関連するコンテンツをすぐに開放します。コンテンツの期限が切れたときに広告が表示されている場合、コンテンツはビューがオフスクリーンに移動したらすぐに解放されます。
- 広告オブジェクトがモーダルの場合、そのコンテンツはユーザが広告を閉じた直後に期限切れになります。

リスト 3-5 (28 ページ) に、広告オブジェクトがそのコンテンツを解放した後で、アプリケーションで実装する必要がある最小限の動作を示します。アプリケーションは追加タスクを実行する必要があります。たとえば、モーダルアプリケーションは、次に広告が必要になったときに表示する広告を保持するために新しい広告オブジェクトを作成します。モードレスアプリケーションは広告の親となっていたビューを削除し、ほかのビューに移動して、コンテンツが表示されていた空白を埋めます。両方の動作については、『*iAdInterstitialSuite*』にサンプルの例が示されています。

リスト 3-5 広告オブジェクトがコンテンツを解放した後にその広告オブジェクトを解放する

```
- (void)interstitialAdDidUnload:(ADInterstitialAd *)interstitialAd
{
    [interstitialAd release];
}
```

フルスクリーン広告エラーの処理

フルスクリーン広告オブジェクトがエラーに遭遇した場合、広告オブジェクトはデリゲートの `interstitialAd:didFailWithError:` メソッドを呼び出して、エラーを処理します。フルスクリーン広告オブジェクトは、`ADBannerView` クラスとエラーコードを共有します。アプリケーションでエ

ラーを表示すべきではありません。エラーが発生した場合の最小限の応答は、広告オブジェクトを解放することです。広告オブジェクトがそのコンテンツを解放した場合のように、ほかのアクションを取ることも可能です。

リスト 3-6 フルスクリーン広告エラーの記録

```
- (void)interstitialAd:(ADInterstitialAd *)interstitialAd  
didFailWithError:(NSError *)error  
{  
    NSLog(@"interstitialAd < %@ > recieved error < %@ >", interstitialAd, error);  
    [interstitialAd release];  
}
```

フルスクリーン広告オブジェクトから戻るほとんどのエラーは同様に処理されます。例外は、フルスクリーン広告オブジェクトが `ADErrorConfigurationError` をアプリケーションに戻す場合です。設定エラーは、iTunes Connectのアプリケーションに対してiAdが有効になっていないことを示します。アプリケーションに広告が表示されるのは、iAdを有効にした後だけです。

iAdアプリケーションのテスト

iAdアプリケーションのテストは重要です。iAdを使いこなすことで、iAdが提供するリッチメディア広告をユーザが容易に探索することができます。そして、アプリケーションによって得られる収入がさらに増えることになります。

バナー広告のテスト

アプリケーションを開発している間、iAd Networkはアプリケーションにテスト広告を送信します。デベロッパの実装検証を支援するため、iAd Networkではアプリケーションのエラー処理コードをテストするためにエラーを返すことがあります。デバイスのワイヤレス機能をオフにしてエラー処理サポートを手作業でテストすることもできます。

iAd Networkは、[表 4-1](#)（31 ページ）に示すように、使用しているテストデバイスにダウンロードされたアプリケーションバイナリに応じて正しい広告を自動的に表示します。

表 4-1 アプリケーションで表示される広告

アプリケーション	対象	表示する広告
デベロッパのビルド	デベロッパ	iAd Networkはテスト広告を提供します。
アドホック配布用ビルド	ベータテスター	iAd Networkはテスト広告を提供します。
署名付き配布用ビルド	エンドユーザ	iAd Network Agreementに署名したデベロッパで、アプリケーションの広告機能を有効にしている場合、iAd Networkはリアルタイムの広告を提供します。

チェックリストのテスト

このチェックリストを使用すると、iAdアプリケーションのテストが容易に行えます。

Banner View

- ユーザインターフェイスに合わせてiAdバナーは向きを変えますか。「[バナーサイズの動的変更](#)」（18 ページ）を参照してください。
- アプリケーションでは広告がロードされたバナーだけが表示されますか。「[広告をロードしたときの応答](#)」（16 ページ）および「[エラー処理](#)」（17 ページ）を参照してください。

- ユーザがバナーをタップしたときにほかのアクティビティが一時停止しますか。 [「Banner Viewでのタッチへの応答」](#)（15 ページ）を参照してください。
- アプリケーションでメモリ不足の警告を監視していますか。「メモリを効率的に使用する」 in *iOS App Programming Guide*を参照してください。
- iAdアクションが終了したら、すべてのサービスは再開しますか。 [「BannerViewでのタッチへの応答」](#)（15 ページ）を参照してください。
- アプリケーションがiAdアクションをキャンセルした場合、すべてのサービスはもう一度再開しますか。 [「広告アクションの取り消し」](#)（18 ページ）を参照してください。
- ユーザがコンテンツの新しい画面に移動した場合、バナービューはすぐに表示されますか。 [「Banner Viewベストプラクティス」](#)（19 ページ）を参照してください。

フルスクリーン広告

- フルスクリーンバナーも方向を変えますか。 [「広告の表示」](#)（25 ページ）を参照してください。
- ユーザがバナーをタップしたときにほかのアクティビティが一時停止しますか。 [「広告アクションの開始」](#)（26 ページ）を参照してください。
- アプリケーションでメモリ不足の警告を監視していますか。「メモリを効率的に使用する」 in *iOS App Programming Guide*を参照してください。
- iAdアクションが終了したら、一時停止していたすべてのサービスは再開しますか。 [「広告アクションの完了」](#)（27 ページ）を参照してください。
- アプリケーションがiAdアクションをキャンセルした場合、一時停止したすべてのサービスは再開しますか。 [「広告アクションの取り消し」](#)（27 ページ）を参照してください。
- ユーザに表示する前にフルスクリーン広告の期限が切れていませんか。 [「広告オブジェクトの作成」](#)（24 ページ）を参照してください。
- ScrollViewに配置されたフルスクリーン広告の期限が切れた場合、アプリケーションはコンテンツビューを正しくクリーンアップしていますか。 [「広告オブジェクトのコンテンツが一定時間しか存続しない場合」](#)（28 ページ）を参照してください。

書類の改訂履歴

この表は「iAdプログラミングガイド」の改訂履歴です。

日付	メモ
2011-09-14	iAdバナーの縦長モードでのサイズを訂正しました。
2011-06-06	バナー広告のベストプラクティスを新たに追加しました。
2011-02-24	iPad上のフルスクリーン広告についての章を追加しました。バナー広告のベストプラクティスを追加しました。
2010-11-15	iOS 4.2用に更新しました。
2010-08-26	iAdがiPadをサポートしたことを追加しました。iAdアプリケーションで期待される動作を明確にしました。
2010-05-27	アプリケーションでの広告の表示方法について説明した新規文書。

改訂履歴

書類の改訂履歴