

# iAd Extension for iOS

Copyright © 2012-2015 Milkman Games, LLC. All rights reserved.

<http://www.milkmangames.com>

For support, contact [support@milkmangames.com](mailto:support@milkmangames.com)

## Before you begin:

**The Iad Extension requires the latest version of Adobe AIR. You can download the latest version of AIR at <http://www.adobe.com/devnet/air/air-sdk-download.html>.**

**To View full AS3 documentation, see 'docs/as3docs/index.html'.**

**Review 'example/IAdMobExample.as' for a sample application.**

**Note that IAdExample.as is a Document Class. If you're a Flash Professional user and don't know how to use a document class, see the FAQ "How do I use the IAdExample.as class in Flash Professional?" at the end of this guide.**

## 1. Set Up iAds in iTunes Connect

You'll need to set up iAd for your apps inside iTunes Connect before you can use the extension.

1. Log into the Apple Developer center at <http://developer.apple.com> and go to the iOS Developer area.
2. Select 'iTunes Connect' from the right hand side.
3. (Before you can use iAd in your apps, you must have the iAd Developer Contract Agreement with Apple digital signed. You can do this by selecting iAd Network from the Agreements, Tax, and Banking section in iTunes Connect, and following on screen instructions.)

**Contracts, Tax, and Banking**

**Request Contracts**

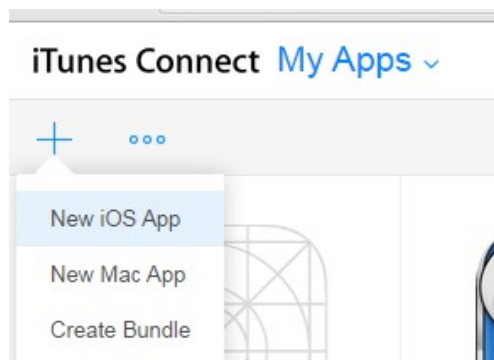
Select the contract(s) you would like to view from the list and click Request. You can distribute your free apps without entering into the contracts below.  
Note: Only users with the Legal role can enter into contracts.

Contract Region	Contract Type	Legal Entity	
World	iAd Network		<a href="#">Request</a>
All	iOS Paid Applications		<a href="#">Request</a>

**Contracts In Effect**

Contract Region	Contract Type	Contract Number	Contact Info	Bank Info	Tax Info	Effective Date	Expiration Date	Download
World	iOS Free Apps		N/A	N/A	N/A	Oct 01, 2010	Oct 01, 2011	N/A

4. Choose 'My Apps', then '+'; 'New iOS App'
- NOTE: if you already have an app set up, you can skip to step 9.**



5. Enter a Name and SKU for your application. Under 'Bundle ID', choose 'Register a new Bundle ID on the Developer Portal', which will bring you to the Provisioning Portal.
6. In the Provisioning Portal, choose 'New App ID'.

7. Enter a description for your new app. For Bundle Identifier, choose a unique name in the recommended scheme (like 'com.yourcompany.YourGameName'). **Take note of what you choose here, as you'll need to match it exactly in your application.xml descriptor file.**

8. Close the pop up Provisioning Portal, and select the Bundle ID you just created back under 'App Information'. If you don't see your new Bundle ID listed, refresh the page, or try going back and pressing 'New App' again. When you're done, press 'Continue'.

9. Continue through the next couple of steps, entering the name, description, keywords, screenshots and etc. for your application.
10. Back on the iTunes Connect screen, select 'iAd' and ensure iAd is active.
11. You're now ready to start building an iAd enabled AIR application!

## ***2. Install the AIR 16.0 (or later) SDK in your IDE***

**The extension requires the AIR 16.0 SDK or later.** You can download the latest AIR SDK from <http://www.adobe.com/devnet/air/air-sdk-download.html>. If you haven't already installed the AIR 16.0 SDK (or higher) for your Flash CS6 or Flash Builder IDE, follow the instructions below:

### **Enabling the AIR 16.0 SDK (or higher) in Flash Professional CS6+:**

1. Unzip the AIR 16.0 SDK (or higher) package to a location on your hard drive.
2. Launch Flash Professional.
3. Select Help > Manage AIR SDK...
4. Press the Plus (+) Button and navigate to the location of the unzipped AIR SDK
5. Press OK
6. Select File > Publish Settings
7. Select the AIR 16.0 (or higher) SDK for iOS from the 'Target' Dropdown menu

### **Enabling the AIR 16.0 SDK (or higher) in Flash Builder 4.6+ on Windows:**

1. Unzip the AIR 16.0 SDK package to a location on your hard drive.
2. Close Flash Builder.
3. Locate the Flash Builder SDK directory. On the PC, this is usually c:\Program Files\Adobe\Adobe Flash Builder 4.6\sdk .
4. Make a copy of the current Flex SDK directory, and give it a descriptive name. For instance, copy the "4.6.0" SDK folder inside /sdk and name the copy "4.6.0\_AIR16".
5. Copy and paste the contents of the AIR 16 SDK on top of the 4.6.0\_AIR16 directory. Accept all changes.
6. Edit the flex-sdk-description.xml file inside the new directory, and change the value of the <name> tag to 'Flex 4.6.0 (AIR 16)'.
7. Open Flash Builder and choose Project > Properties > Flex Compiler > Configure Flex SDKs.
8. Press 'Add' and navigate to the new folder location.

### **Enabling the AIR 16 SDK (or higher) in Flash Builder 4.6+ on a Mac:**

1. Copy the contents AIR 16 SDK package to a location on your hard drive.
2. Close Flash Builder.
3. Locate the Flash Builder SDK directory. On the Mac, it is usually /Applications/Adobe Flash Builder 4.6/sdk/. On the PC, c:\Program Files\Adobe\Adobe Flash Builder 4.6\sdk .
4. Create a new folder inside the SDK folder, called AIR16SDK and copy the contents of the SDK package into it.
5. Open the Terminal, and merge the AIR 16 SDK files into your current SDK directory:

```
sudo cp -Rp /Applications/Adobe\ Flash\ Builder\ 4.6/sdks/AIR35SDK/ /Applications/Adobe\ Flash\ Builder\ 4.6/sdks/4.6.0/
```

6. Edit the flex-sdk-description.xml file inside the new directory, and change the value of the <name> tag to 'Flex 4.6.0 (AIR 16)'.

7. Open Flash Builder and choose Project > Properties > Flex Compiler > Configure Flex SDKs.

Press 'Add' and navigate to the new folder location.

## 2. Include the ANE Library

Add the Iad ANE library to your project.

### In Flash Professional:

1. Create a new mobile project
2. Choose File>Publish Settings...
3. Select the wrench icon next to 'Script' for 'ActionScript Settings'
4. Select the Library Path tab.
5. Click 'Browse for Native Extension (ANE) File' and select the com.milkmangames.extensions.IAd.ane file. Press OK.
6. Press OK

### In Flash Builder 4.6 or higher:

1. Go to *Project Properties*
2. Select *Native Extensions* under *Actionscript Build Path*
3. Choose *Add ANE...* and navigate to the com.milkmangames.extensions.IAd.ane file
4. Select *Actionscript Build Packaging > iOS*
5. Select the *Native Extensions* tab, and click the '*Package*' checkbox next to the extension

## 3. Add Actionscript Code

The iAd extension can be up and running in a few simple calls. See '**example/IAdExample.as**' for a full example, including creating and destroying ad,s setting visibility, error capturing, and more.

## Initialize the Iad Extension

1. Import the API Classes:

```
import com.milkmangames.nativeextensions.ios.*;
import com.milkmangames.nativeextensions.ios.events.*;
```

2. Initialize the API by calling IAd.create().

You'll also want to check that the current platform is supported by the extension before initializing (for instance, IAd won't work on Android):

```
if(IAd.isSupported())
{
    IAd.create();
}
```

```

else
{
    trace("IAd won't work on this platform!");
    return;
}

```

## Create and Display a Banner Ad

Create a Banner Ad container by calling `createBannerAd()`.

The extension will automatically handle the lifecycle of banner ads from here on out; automatically trying to get a new ad after one fails to load, automatically showing a new ad when its ready, and automatically hiding the ad view when there's a problem – all in accordance with Apple's requirements.

This example code creates a banner ad container, at the top of the screen, that can show either portrait or landscape content.

```

// create a banner ad container at the top of the screen
IAd.iAd.createBannerAd(IAdBannerAlignment.TOP, IAdContentSize.PORTRAIT_AND_LANDSCAPE);

```

## Optional: Loading Interstitial Ads

Interstitial ads are a special, full screen ad type that can produce higher revenue for your application than traditional banners.

To display an interstitial ad, call the `loadInterstitial()` function. The 'true' parameter tells the extension to show the interstitial automatically as soon as it's loaded.

```

// load interstitial, and show it when its ready
IAd.iAd.loadInterstitial(true);

```

Interstitial ads are only available on iPads, and iPhones and iPod Touches running iOS 7 or higher. Optionally, you can check if they're available on the current device with the `isInterstitialAvailable()` function.

```

// check if this is an ipad or ios7+ phone and can therefore show an interstitial
if(IAd.iAd.isInterstitialAvailable())
{
    trace("Interstitials will work here.");
}
else
{
    trace("Sorry, interstitials only work on iPads.");
}

```

**Important Note:** You cannot load a new interstitial if the last interstitial has not yet finished- instead of loading interstitials in rapid succession, wait for the previous interstitial to dispatch `INTERSTITIAL_DISMISSED` or `INTERSTITIAL_AD_FAILED` first (see "Optional: Add Event Listeners below."):

```

try
{
    IAd.iAd.loadInterstitial(true);
}
catch (e:Error)
{
    trace("Can't load right now- did you wait for the old interstitial to finish?");
}

```

## Optional: Preloading Interstitial Ads

Optionally, you can begin loading an interstitial at the beginning of your app, but not display it until later. This is useful for giving the ad time to load, so it's ready to display during a quick transition such as a game over screen.

By setting the parameter of `loadInterstitial` to `false`, the extension will not automatically show the ad when it's loaded, but instead hold it in memory:

```
// preload interstitial, but don't show it until you're ready
IAd.loadInterstitial(false);
```

When the preload is complete, an `IAdEvent.INTERSTITIAL_AD_LOADED_AD` event will be dispatched:

```
IAd.iAd.addEventListener(IAdEvent.INTERSTITIAL_AD_LOADED,onInterstitialAdLoaded);
```

```
function onInterstitialAdLoaded(e:IAdEvent):void
{
    trace("Interstitial has loaded");
}
```

You can then use the `showPendingInterstitial()` function to display the preloaded ad. (You may also check the `IAd.isInterstitialReady()` function to check if a preload is complete at any time):

```
if(IAd.iAd.isInterstitialReady())
{
    IAd.iAd.showPendingInterstitial();
}
```

**Important Note:** You cannot pre-load a new interstitial if the last interstitial has not yet finished- instead of loading interstitials in rapid succession, wait for the previous interstitial to dispatch `INTERSTITIAL_DISMISSED` or `INTERSTITIAL_AD_FAILED` first (see "Optional: Add Event Listeners below."):

```
try
{
    IAd.iAd.loadInterstitial(false);
}
catch (e:Error)
{
    trace("Can't preload again yet- did you wait for the old interstitial to finish?");
}
```

## Optional: Hiding or Destroying Ads

You may wish to remove a banner ad from the screen- either by destroying it entirely, or temporarily making it invisible.

For instance, if your application is switching to a sub menu momentarily, and you do not want the ad to display, you might set it be invisible while the menu shows, and then visible again when the menu is complete.

```
// hide an ad temporarily
IAd.iAd.setBannerVisibility(false);

// to show it again...
IAd.iAd.setBannerVisibility(true);
```

By default, the appearance or disappearance of the banner will instantaneous. If you want to animate the transition smoothly, add a second parameter and set it to `true`:

```
// hide an ad, with a smooth transition
IAd.iAd.setBannerVisibility(false,true);
```

If your application is switching to a mode where the ad will not be seen again for a long time, or ever, you may

prefer to destroy the ad entirely. In this case, the ad will stop making requests so you can avoid wasting memory and ad inventory:

```
// destroy an ad entirely.  
iAd.iAd.destroyBannerAd();
```

## Optional: Add Event Listeners

All event listeners are optional; however, they can be very useful for debugging if you're having trouble loading ads, or to fine tune your app's behavior when ads change state.

1. When the user clicks on a Banner Ad, one of two things might happen. Either they will be taken out of your application, to view the App Store or Safari, or they will be shown a full screen pop up ad. In the case of a pop-up ad, you'll want to stop sounds in your application and pause any major actions. You can tell whether the user is going to be viewing a pop up or leaving the application by the events *willLeaveApplication* property:

```
iAd.iAd.addListener(iAdEvent.BANNER_ACTION_BEGIN, onBannerActionBegin);
```

```
function onBannerActionBegin(e:iAdEvent):void  
{  
    trace("the user started interacting with an ad.");  
    if(e.willLeaveApplication)  
    {  
        // the app is about to exit to go somewhere else.  
    } else  
    {  
        // you might want to pause sounds here, etc.  
    }  
}
```

2. When the user is done with the banner ad, they may be returned to your application. If you had paused actions and sounds in the previous event, you'll want to resume them when the interaction is over:

```
iAd.iAd.addListener(iAdEvent.BANNER_ACTION_FINISHED, onBannerActionFinished);
```

```
function onBannerActionFinished(e:iAdEvent):void  
{  
    trace("the user finished interacting with an ad.");  
    // you might want to resume sounds here, etc.  
}
```

3. Interstitials dispatch slightly different events- *iAdEvent.INTERSTITIAL\_SHOWN* when an ad appears on the screen, and *iAdEvent.INTERSTITIAL\_DISMISSED* when it is removed from the screen. You may need to track these events to pause and unpaue your application while an interstitial is shown.
4. Additionally, you may want to listen for *iAdEvent.INTERSTITIAL\_AD\_UNLOADED* and *iAdEvent.INTERSTITIAL\_AD\_LOADED*, and *iAdErrorEvent.INTERSTITIAL\_AD\_FAILED* events to track the loading and unloading of interstitial views:

```
// listen for events  
iAd.iAd.addListener(iAdEvent.INTERSTITIAL_AD_LOADED, onInterstitialAdLoaded);  
iAd.iAd.addListener(iAdEvent.INTERSTITIAL_AD_UNLOADED, onInterstitialAdUnloaded);  
iAd.iAd.addListener(iAdErrorEvent.INTERSTITIAL_AD_FAILED, onInterstitialAdFailed);
```

5. If iAd succeeds in loading an interstitial, you'll receive the *iAdEvent.INTERSTITIAL\_AD\_LOADED* event; if you loaded the ad with *autoshow* set to true, your screen will now be covered completely by the ad. If there's an error and iAd can't get an interstitial ad, you'll received

*IALErrorEvent.INTERSTITIAL\_AD\_FAILED*. Finally, the *IAAdEvent.INTERSTITIAL\_AD\_UNLOADED* event will happen when the ad is being removed from the memory.

```
function onInterstitialAdLoaded(e:IAAdEvent):void
{
    trace("the interstitial loaded and will show now, if loaded with autoShow true.");
}

function onInterstitialAdUnloaded(e:IAAdEvent):void
{
    trace("the interstitial has been unloaded..");
}

function onInterstitialAdFailed(e:IALErrorEvent):void
{
    trace("there was an error loading the ad.");
}
```

## 4. Update Your Application Descriptor

Include the extension in your Application Descriptor XML. For a working example, see '**example/app.xml**'. If you're using Flash or Flash Builder, these settings may have already been made for you.

Include a link to the extension in the descriptor:

```
<extensions>
<extensionID>com.milkmangames.extensions.IAd</extensionID>
</extensions>
```

## 4. Build the Application

### Flash Professional- Windows:

Flash CS6 has full support for native extensions.

1. Make sure you've included the .swc file as described in Section 2, '*Include the Iad ANE Library.*'
2. Compile your iOS application as usual.

### Flash Builder:

1. Make sure you've included the ANE file as described in Section 2, '*Include the Iad ANE Library.*'
2. Make sure that the extension is enabled for your target configuration.
3. Build your application and deploy as usual.

## 5. Testing Vs Release

1. **When testing:** Be sure the .mobileprovision file you built the ipa with was your **development** mobileprovision. (but you can and should use the distribution mobileprovision when building your final app to upload to the App Store.)
2. **When releasing to the store:** Be sure the .mobileprovision file you built the ipa with was your **release** mobileprovision.



## 6. Troubleshooting

### Why aren't ads displaying?

- Make sure you've agreed to the iAds contract with Apple in the Contracts section.
- During testing, Apple will send both 'successful' and 'failed' ad requests to simulate a situation where inventory is not available or the network fails. Wait for another ad to load, and monitor the AD\_LOADED / AD\_FAILED events as described above.
- If you're adding iAds to an already published game, you need to create a new Version, and enable iAds on that. Then update the version number appropriately in your new version's application.xml descriptor file.

### Why does the test fullscreen start in portrait, even though my app is landscape?

- The Test Ad may start in portrait, depending on the device. In production, the appropriate ads will always be shown however, as long as you've enabled the correct IAdContentSize settings when calling createBannerAd().

### Why did my app crashed when I called createBannerAd?

- For ads to work, you must set the appropriate content size for your ads orientation; if your app is portrait, for instance, portrait must be specified. The opposite goes for landscape. Refer to section 3.4 for information on setting the IAdContentSize parameter.

### Why don't interstitials show up on my iPhone?

- Interstitials only display on iPhones that are running iOS 7 or higher.

### How do I use the IAdExample.as file in Flash Professional?

- First, setup iTunes Connect, create the application and add the extension by following this guide, Sections 1-2.
- Copy and paste IadExample.as file into the same folder as your .fla. Do not copy and paste its contents on to the timeline. That will not work.
- In Flash properties, under 'Document Class', type 'IAdMobExample' (no quotes) and press OK.
- Build and install the application.

### I'm not a programmer, what's some minimal code I can copy and paste on to my Flash timeline on Frame 1 to make this work?

The code below will work for that. You still need to set up your application on iTunes Connect and set up the extension in Flash though (Sections 1-2 above!):

```
import com.milkmangames.nativeextensions.ios.*;
import com.milkmangames.nativeextensions.ios.events.*;

if(IAd.isSupported())
{
    IAd.create();

    IAd.iAd.createBannerAd(IAdBannerAlignment.TOP,IAdContentSize.PORTRAIT_AND_LANDSCAPE);
}
```

### Why doesn't the extension work when I run my swf on my Mac / BlackBerry / Windows Computer?

- The Iad extension relies on functions of the mobile operating system for iOS. The extension will only work when you run it on an iOS phone or tablet.

### **I still need help! How can I get technical support?**

- Your purchase comes with free technical support by email – just send us a message at [support@milkmandgames.com](mailto:support@milkmandgames.com) . We're here to help! Please remember that...:
- We're open during United States business hours, excluding U.S. Holidays, Monday-Friday, Pacific Standard Time. We strive to answer all support email within 24 hours (not including weekends and holidays) but usually do so much faster. Remember that we may not be in the same time zone.
- Please remember to mention: which extension you're having a problem with, what IDE you're using (such as 'Flash CS6' or 'Flash Builder 4.6'), and what device you're targeting. If you're experiencing an error message, please specify what that message is.
- We don't provide tech support through blog comments, Facebook, or Twitter. Please email us and we'll be happy to help you out!