User interface design

for iOS, using Objective-C

# Lab 4 – iAds

## Functional Goals

Take a finished iOS app and monetize it using iAds.

## Learning Goals

* Understand iOS paradigms, including the iAd APIs.

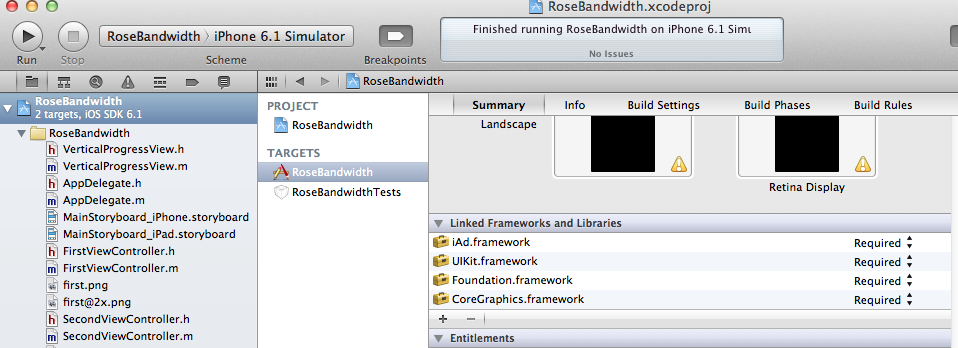
## Prerequisites

* You’ll need to install Xcode (Version 4.5.2 was used to create this lab) from the App Store
* A basic understanding of Xcode and Objective C development
* Code from Lab 3 (a complete version may be available from your instructor)

## Submission Instructions

Submit answers to the **3** (or **4**, with extra credit)questions in this lab as a .pdf to the appropriate Moodle submission form.

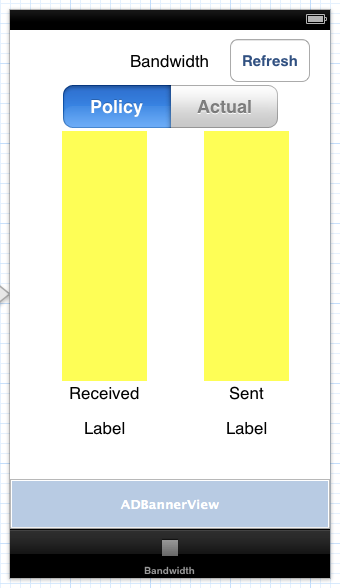
## Get started: Add the iAd framework



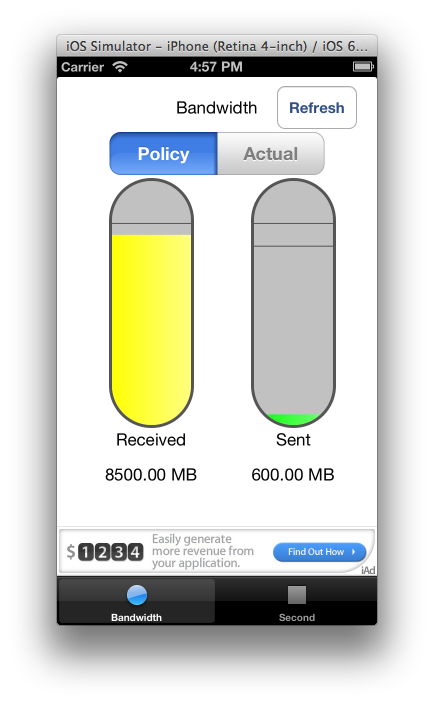
Go into your RoseBandwidth target and add a new Linked Framework by pressing the + button and selecting iAd.framework.

## Add the iAd in your Storyboard!

Head to your iPhone storyboard and put an AdBannerView at the very bottom of your view. You may need to shift things around a bit to make it fit, this is fine.

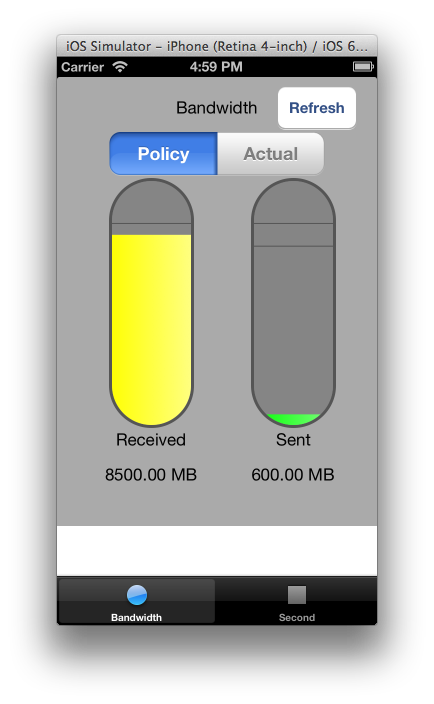


Go ahead and build and run your app – you should see some sample ads showing up on your screen! It was that easy!



## Making the view appear and disappear when Ads are present

Right now we have a white background that is masking something. When there is no ad present, the AdBannerView shows up white. If we had a grey background, as below, it would look extremely bad when an ad wasn’t loaded.



How do we fix this? First we need a reference to our ad banner view. Create the property and hook it up in Interface Builder. You should be able to do this with ease at this point.

@property (nonatomic, strong) IBOutlet ADBannerView \*adView;

Make sure to synthesize this in FirstViewController.m as well.

In order to interact with our ad, we need to be come an AdBannerViewDelegate. We do this by changing the FirstViewController declaration to this:

@interface FirstViewController : UIViewController <BandwidthScraperDelegate, ADBannerViewDelegate>

In our viewDidLoad method we want to do 2 things – initialize it to hidden, and set ourselves as the delegate.

self.adView.delegate = self;

adView.hidden = YES;

Now we simply implement the following methods to show the ad when one is loaded and hide it otherwise!

- (void)bannerViewDidLoadAd:(ADBannerView \*)banner

{

adView.hidden = NO;

}

-(void)bannerView:(ADBannerView \*)banner didFailToReceiveAdWithError:(NSError \*)error

{

adView.hidden = YES;

}

## Congratulations

You’re done; run your app and check it out!