

# AdMob Mediation iOS Ad Network Adapter API Change Note

Mar 29, 2012

[AdMob Mediation iOS Ad Network Adapter API Change Note](#)

[Requesting Ads from Adapters](#)

[Report Clicks Explicitly](#)

[Ad Network Extras](#)

[Presenting Interstitials](#)

[Adapter Version](#)

[Removed containerViewBounds](#)

We have made some changes to the GADAdNetworkAdapter protocol. Existing ad network adapters **\*must be updated\*** in order to work with the new AdMob Mediation SDK. This document describes the changes, and what you need to do to update your Ad Network Adapter code.

In brief, these are the changes:

1. Replaced -getAdWithType: with two methods, -getBannerWithSize: and -getInterstitial that Ad Network Adapters must implement. Introduced GADAdSize type for banner ads and removed the GADAdType enum.
2. Requiring adapter:clickDidOccurInBanner: callback for clicks.
3. Changed the mechanism for publishers to pass additional information to the ad network adapters. (Implement the GADAdNetworkExtras protocol.)
4. -presentInterstitialFromRootViewController: is now a required method.
5. +adapterVersion is now a class method.
6. Removed the method -containerViewBounds from the GADAdNetworkConnector protocol.

## Requesting Ads from Adapters

You must split the -getAdWithType method into two methods, -getBannerWithSize: and -getInterstitial. After the code is moved, there is no need for -getAdWithType: any more, so it can be removed.

We have also removed the GADAdType enum and replaced it with the GADAdSize type.

In your -getBannerWithSize: method, you should check whether the ad size is a size that you support, then create an ad that matches the supplied ad size. You should use convenience functions in GADAdSize.h to operate on GADAdSize. Treat the type as opaque. Do not access any fields directly.

To illustrate the necessary change, here is an skeleton example of -getAdWithType:. The bolded code represent code that needs to be moved.

```

- (void)getAdWithType:(GADMAType)type {
    if (type == kGADMATypeInterstitial) {
        // Request interstitial.
    } else {
        CGSize cgAdSize;
        switch (type) {
            case kGADMATypeBanner:
                cgAdSize = CGSizemake(320, 50);
                break;
            case kGADMATypeMediumRectangle:
                cgAdSize = CGSizemake(300, 250);
                break;
            case kGADMATypeFullBanner:
                cgAdSize = CGSizemake(468, 60);
                break;
            case kGADMATypeLeaderboard:
                cgAdSize = CGSizemake(728, 90);
                break;
            case kGADMATypeSkyscraper:
                cgAdSize = CGSizemake(120, 600);
                break;
            default:
                // Unrecognized ad size. Call back with error.
                [connector adapter:self didFailAd:error];
                return;
        }
        // Create ad with a size of cgAdSize
    }
}

```

The split methods will look like:

```

- (void)getInterstitial {
    // Request interstitial.
}

- (void)getBannerWithSize:(GADAdSize)adSize {
    if (!GADAdSizeEqualToSize(adSize, kGADAdSizeBanner) &&
        !GADAdSizeEqualToSize(adSize, kGADAdSizeFullBanner) &&
        !GADAdSizeEqualToSize(adSize, kGADAdSizeLeaderboard) &&
        !GADAdSizeEqualToSize(adSize, kGADAdSizeMediatedLeaderboard) &&
        !GADAdSizeEqualToSize(adSize, kGADAdSizeMediumRectangle) &&
        !GADAdSizeEqualToSize(adSize, kGADAdSizeSkyscraper)) {
        // Unrecognized ad size. Call back with error.
        [connector adapter:self didFailAd:error];
    }
    CGSize cgAdSize = CGSizeFromGADAdSize(adSize);
}

```

```
// Create ad with a size of cgAdSize
}
```

## Report Clicks Explicitly

We now require explicit callbacks when the user clicks. You should no longer assume that the Mediation SDK infer clicks in the `adapterWillPresentFullScreenModal:` callback method. When the user taps your ad, please call `adapter:clickDidOccurInBanner:` of the Ad Network Connector. Calling this is important for correct reporting for publishers.

## Ad Network Extras

If you need information beyond what is available in `GADAdNetworkConnector`, you may use the Ad Network Extras mechanism. The old method is to set a dictionary under `additionalParameters` of `GADRequest`. This may be error-prone and not type safe.

This is the new mechanism:

1. Define your own class that implements the `GADAdNetworkExtras` protocol for publishers to pass any information to you.
2. In your class, you may define any field of any appropriate type.
3. Your ad network adapter must return the class that implements the `GADAdNetworkExtras` protocol from the `+networkExtrasClass` class method.
4. A publisher will instantiate such class and fill in the required information.
5. The publisher then registers it with the AdMob SDK using the `registerAdNetworkExtras:` method of `GADRequest`.
6. Your ad network adapter may then retrieve the object using the `-networkExtras` method of the connector.

Remember to distribute the definition header file to your publishers for them to use.

## Presenting Interstitials

The method `-presentInterstitialFromRootViewController:` is now a required method, even if your ad network does not support interstitials. If your ad network does not support interstitials, implement the method and do nothing inside the method.

## Adapter Version

`adapterVersion` used to be an instance method. It is now a class method. This way we can ask for your adapter's version without having to instantiate an adapter object.

## Removed containerViewBounds

We remove the `-containerViewBounds` method from `GADAdNetworkConnector`, as the banner view bounds will always be the `CGSize` of the ad size passed to `getBannerWithSize:`. So this method is no longer necessary.