# Android | MoPub - InMobi Unity Integration

Step 1: Configuring InMobi as an Ad Source on MoPub's Dashboard

Log on to your MoPub account and identify the ad units that you wish to monetize via InMobi. Then, navigate to the Networks tab.

I. To monetize via InMobi, select Custom SDK Network. It allows you to integrate InMobi via MoPub with plug-n-play "Custom Events".
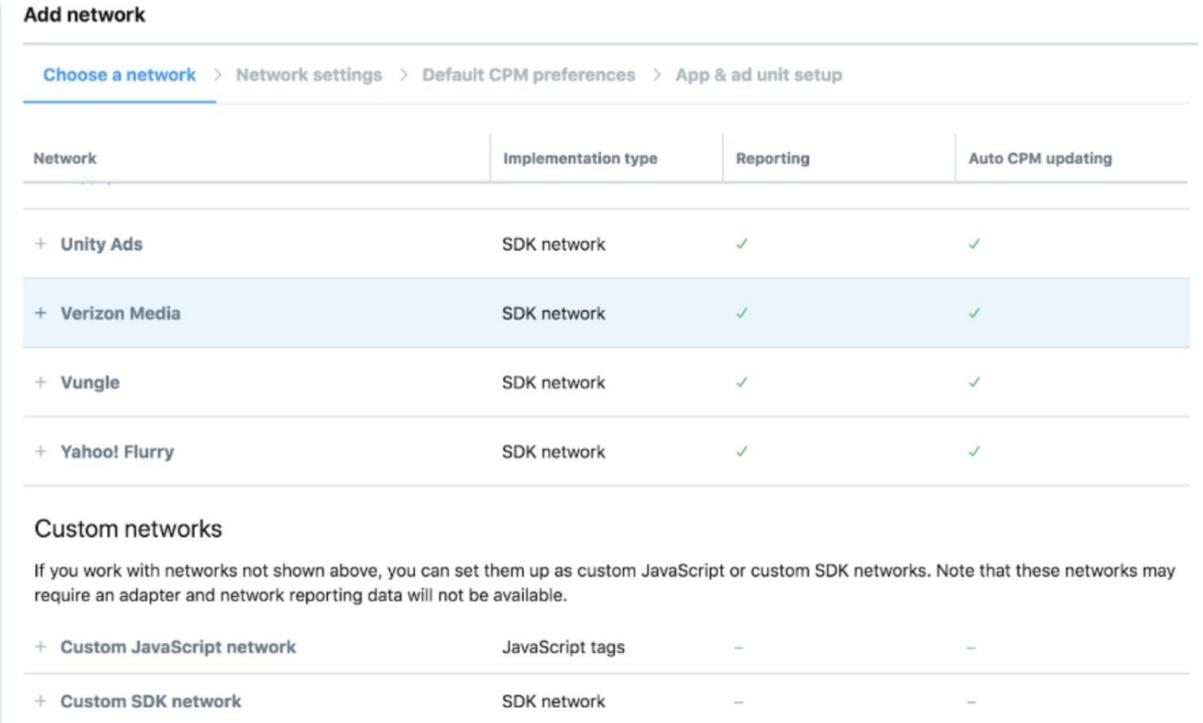
Add network

Choose a network > Network settings > Default CPM preferences > App & ad unit setup

Network	Implementation type	Reporting	Auto CPM updating
+ Unity Ads	SDK network	✓	✓
+ Verizon Media	SDK network	✓	✓
+ Vungle	SDK network	✓	✓
+ Yahoo! Flurry	SDK network	✓	✓

Custom networks

If you work with networks not shown above, you can set them up as custom JavaScript or custom SDK networks. Note that these networks may require an adapter and network reporting data will not be available.

+ Custom JavaScript network	JavaScript tags	–	–
+ Custom SDK network	SDK network	–	–

II. Navigate to the specific ad unit that you want to monetize via InMobi.

III. Enter the name of InMobi custom event class specific for that particular ad format as Custom Event Class value. For example, Interstitial Ad - the custom event is called: **"com.inmobi.showcase.InMobiInterstitialCustomEvent".**
Use fully qualified Custom Event class names. Here is the complete grid for custom events by ad unit type:

Ad Unit	Class Name
Banner	`com.inmobi.showcase.InMobiBannerCustomEvent`
Medium (300x250 banner)	`com.inmobi.showcase.InMobiBannerCustomEvent`
Fullscreen	`com.inmobi.showcase.InMobiInterstitialCustomEvent`
Rewarded Video	`com.inmobi.showcase.InMobiRewardedCustomEvent`
Native	`com.inmobi.showcase.InMobiNativeCustomEvent/` `com.inmobi.showcase.InMobiNativeStrandCustomEvent`

IMPORTANT: On **Unity**, please **DO NOT** change the InMobi class name.

IV: Enter the Account ID and the Placement ID in the following format in the placeholder for Custom Event Class Data:
{"placementid":"<Enter your InMobi Placement ID here>","accountid":"<Enter your InMobi Account ID here>"}

A sample input will look like this:
{"placementid":"1468850508657","accountid":"2072131215e04a5ea20e 25ffec1a7171"}

Step 2: Adding the SDK, Custom Event(as jar) and dependencies to your Project

InMobi SDK
Download the latest version of **InMobi's SD**K [here](#) and add it to your Unity project under the location: **[UNITY PROJECT ROOT]/Assets/Plugins/Android/ .** For ease of maintenance, it's recommend keeping the custom event classes and the network SDKs as separate .jar files
InMobi SDK: https://www.inmobi.com/sdk

Custom Events As JAR
Add the custom events (as .jar files)(Please see attached **JAR**) to **[UNITY PROJECT ROOT]/Assets/Plugins/Android/**.

Picasso:
The InMobi SDK for Android uses the popular Picasso library for loading the ad assets. Download the Picasso 2.71828 [here](#) and add it to your Unity project under the location: **[UNITY PROJECT ROOT]/Assets/Plugins/Android/**

Picasso 2.71828:
https://mvnrepository.com/artifact/com.squareup.picasso/picasso/2.71828

RecyclerView:
For supporting free scroll of a deck of images or ads, you must use RecyclerView. To do so, Download the Recyclerview 28.0.0 here and add it to your Unity project under the location: **[UNITY PROJECT ROOT]/Assets/Plugins/Android/**

Recyclerview:
https://mvnrepository.com/artifact/com.android.support/recyclerview-v7/28.0.0

Note: Failure to include RecyclerView dependency in your application gradle scripts will cause interstitial ad requests to fail, thus affecting monetization of your app with the InMobi SDK.

Chrome Custom Tab:
This is required to redirect the users to URLs outside InMobi WebView. To add the Chrome Custom library to your application:
Download the Chrome Custom Tab here and add it to your Unity project under the location: **[UNITY PROJECT ROOT]/Assets/Plugins/Android/**

Chrome Custom Tab:
https://mvnrepository.com/artifact/com.android.support/customtabs/28.0.0-alpha1
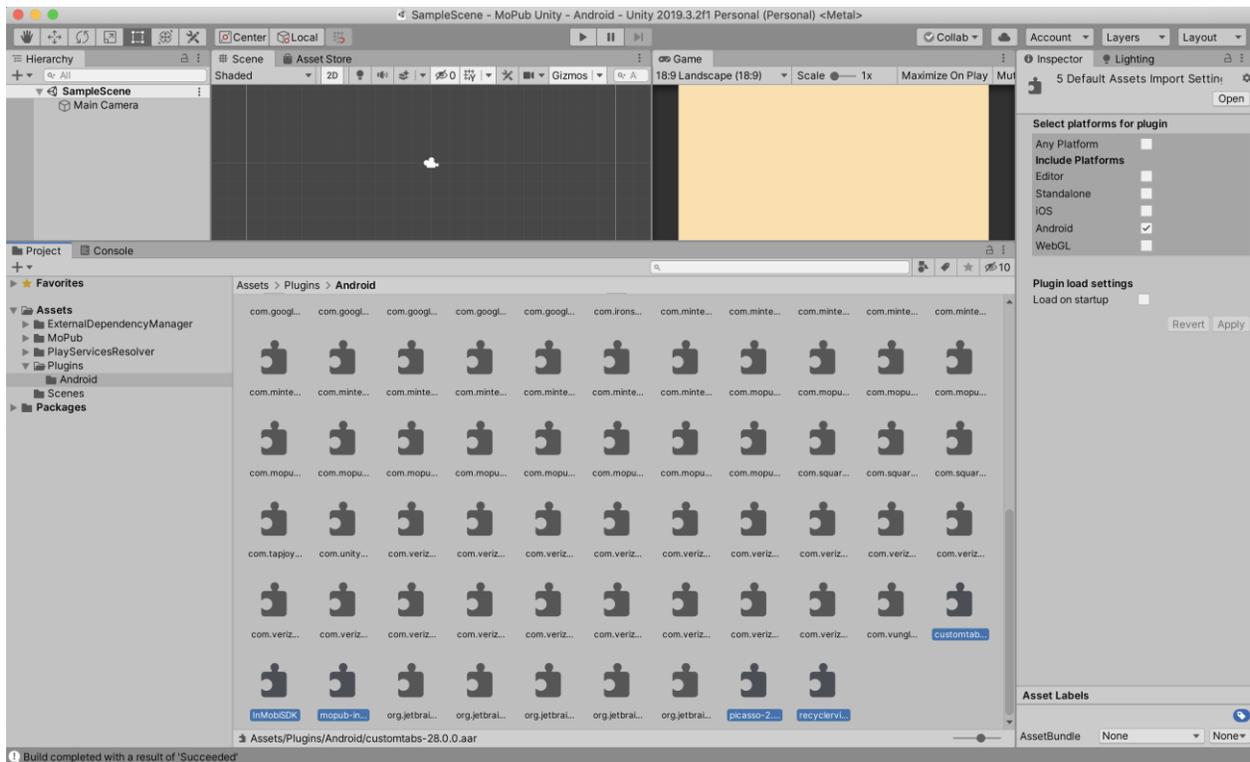
Download links:

InMobi SDK - https://support.inmobi.com/monetize/download-sdk

MoPub-InMobi Adapter -
https://www.dropbox.com/s/ice532569x2dhfx/Mopub%20Unity%20Jar%20.zip?dl=0

Chrome Custom Tabs:
https://maven.google.com/com/android/support/customtabs/28.0.0/customtabs-28.0.0.aar

Picasso: https://repo1.maven.org/maven2/com/squareup/picasso/picasso/2.71828/picasso-2.71828.aar

RecylerView: https://maven.google.com/com/android/support/recyclerview-v7/28.0.0/recyclerview-v7-28.0.0.aar

STEP 3: Integration Testing

Follow the below instructions to ensure MoPub SDK relays the ad request to InMobi for testing:

I. Set manual Ad network optimization for every network including MoPub.

II. Set manual eCPM floors for all networks keeping the eCPM floor for InMobi higher than other networks, this will ensure that the ad request gets relayed to InMobi for testing. For example for InMobi set the floor to $5 while MoPub marketplace is set to $0.05.

III. Configure the test mode on InMobi portal.
Go to Tools - Diagnostics and switch Test Mode to either Global ON or Selective ON.

If you are integrating an ad unit for the first time	Set **Test Mode** to **Global ON**.

If you want to selectively turn on test traffic for a set of devices:
 	
Note: You already have a prior version of the SDK integrated for this particular ad unit and therefore you should restrict your testing to only few devices
	Set **Test Mode** to **Selective ON**.
	In the device section:
	
	  a. In the **Device ID** box, type the device ID.
	  b. In the **Device Name** box, set any name.
	  c. Click **Add Device** to add the test device.
	If you already have a device configured, you can select the device and test mode will be enabled.