



appnexus

28 west 23rd street, floor 4, new york, ny 10010
www.appnexus.com

AppNexus Open AdStream Mobile SDK Client-side Mediation User Guide July 27th, 2015



P (646) 825-6460

F (646) 825-6465

info@appnexus.com



appnexus



appnexus

28 west 23rd street, floor 4, new york, ny 10010
www.appnexus.com

Contents

Overview.....	3
Pre-requisites for Mediation.....	3
Steps to Enable Mediation using OAS Mobile SDK	4
Enabling mediation	21
Setting Targeting parameters for Mediated Banners (Optional).....	25
Setting Targeting parameters for Mediated Interstitials (Optional).....	25
Setting Location Targeting parameters for Mediated Ads (Optional)	26
Enabling Mediation Dynamically	27
Mediation Libraries with Adapters	27
Debugging Mediation	28

P (646) 825-6460

F (646) 825-6465

info@appnexus.com



appnexus



appnexus

28 west 23rd street, floor 4, new york, ny 10010
www.appnexus.com

Overview

Client-side mediation allows publishers to fill remnant ad spots with ads served from AppNexus, as well as multiple 3rd party ad networks. It is initiated by OAS Mobile SDK, which calls AppNexus Mobile SDK to either fetch an ad from AppNexus inventory, or calls out one or more mediated networks/SDKs in a “waterfall”-like process. If OAS Mobile SDK can’t retrieve an ad from or through OAS, it will attempt fetching the ad through AppNexus Mobile SDK and iterate over the list of mediated SDKs in the appropriate order. This is repeated until the ad spot is either filled, or none of the mediated network yields an ad.

Reasons to mediate to another SDK include the following:

- To provide better monetization for remnant ad spots.
- To provide access to information such as a unique user ID or the device’s operating system, location, or ID.
- Some networks only accept requests from their own SDKs, forcing the app developer to use the network’s SDK to access their demand. Mediation through such network’s SDK helps overcoming this limitation.

Pre-requisites for Mediation

1. Publisher application.
2. OAS Mobile SDK version 2.0.0 and above for Android.
3. Supported AppNexus Mobile SDK version 2.1 and above for Android.
4. Supported Mediation libraries and adapters.
5. Supported Mediation libraries and adapters including Amazon, Mopub, Facebook, Millennial Media and Google Play and their corresponding adapters.

Please Note: Millennial Media SDK can be downloaded separately after setting up account with them.





appnexus

28 west 23rd street, floor 4, new york, ny 10010
www.appnexus.com

Steps to Enable Mediation using OAS Mobile SDK

To enable client-side mediation in the application, publisher must integrate AppNexusOASMobileAdSDK, AppNexusSDK, mediation libraries and adapters including Google Play service library.

Please note:

- All the mediation libraries and adapters including AppNexusSDK are optional and should be integrated only when publisher wants to enable mediation.
- Each SDK is linked with the main SDK through its adapter. If publisher is including any particular SDK (for e.g. mopub-sdk) then he must include its corresponding adapter (appnexus-mopub-adapter).

SDK Integration

Please follow these steps to integrate all the required libraries for mediation using OAS Mobile SDK for Android.

There are two ways to integrate the OAS Mobile SDK (with optional mediation libraries) into your app:

1. Integrate the SDK via Maven directly from the app's build.gradle file.
2. Get the latest version of SDK from the GitHub page.

1) Integration via Maven

This method is suitable for Gradle based development environment. This way of integration is comparatively simple and hassle free.

Application developers can easily update their project's build.gradle file as shown below. Once it is updated, it is ready to build and run the app.





appnexus

28 west 23rd street, floor 4, new york, ny 10010
www.appnexus.com

```
repositories {
    mavenCentral()
    maven{
        url 'http://cdn.adnxs.com/anx-sdk/maven'
    }
}
...
dependencies {

    // Integrate OAS SDK and Android support library
    compile 'com.appnexus.oas.mobilesdk:appnexusoasmobileadsdk:2.2.1'
    compile 'com.android.support:support-v4:22.2.0'

    // Optionally Integrate AppNexus SDK
    compile 'com.appnexus.opensdk:appnexus-sdk:2.3'

    // Optionally Integrate Mediation Adapters
    compile 'com.appnexus.opensdk.mediatedviews:appnexus-facebook-mediation:2.2.0-4.1.2'
    compile 'com.appnexus.opensdk.mediatedviews:appnexus-amazon-mediation:2.2.0-5.5.149'
    compile 'com.appnexus.opensdk.mediatedviews:appnexus-googleplay-mediation:2.2.0-7.3.0'
    compile 'com.appnexus.opensdk.mediatedviews:appnexus-mopub-mediation:2.2.0-3.3.0'
    compile 'com.appnexus.opensdk.mediatedviews:appnexus-millennialmedia-mediation:2.2.0-5.3.0'
    compile 'com.appnexus.opensdk.mediatedviews:appnexus-chartboost-mediation:2.2.0-5.2.0'
    compile 'com.appnexus.opensdk.mediatedviews:appnexus-inmobi-mediation:2.2.0-4.5.3'
    compile 'com.appnexus.opensdk.mediatedviews:appnexus-vidoomia-mediation:2.2.0-lw'
    ...
}
```

Please note:

The AppNexus SDK and mediation adapters declaration as mentioned above in the Gradle dependencies, are optional.





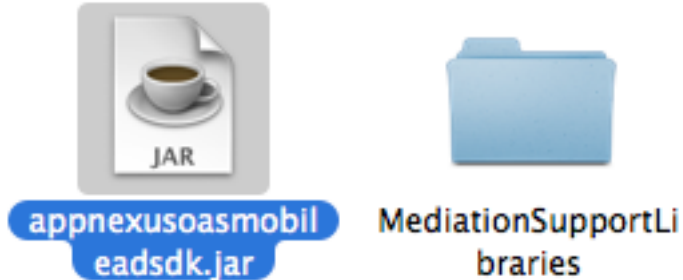
appnexus

28 west 23rd street, floor 4, new york, ny 10010
www.appnexus.com

2) Getting the SDK from GitHub page

AppNexus Open AdStream Mobile SDK zip file (Android_SDK.zip) can be downloaded from our [GitHub](#) page. Please follow the integration steps as shown below.

- **Integrate appnexusoasmobileadsdk.jar.**
- Extract the Android SDK zip file and copy the appnexusoasmobileadsdk.jar from the extracted folder.



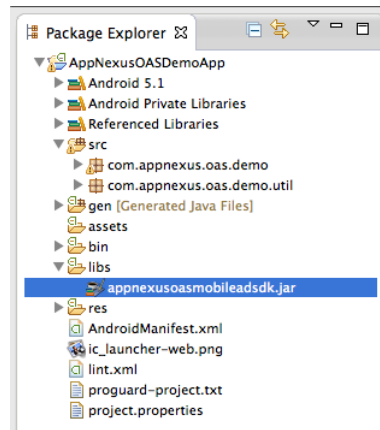
- Paste it in the libs folder of your application project.



appnexus

www.appnexus.com

28 west 23rd street, floor 4, new york, ny 10010

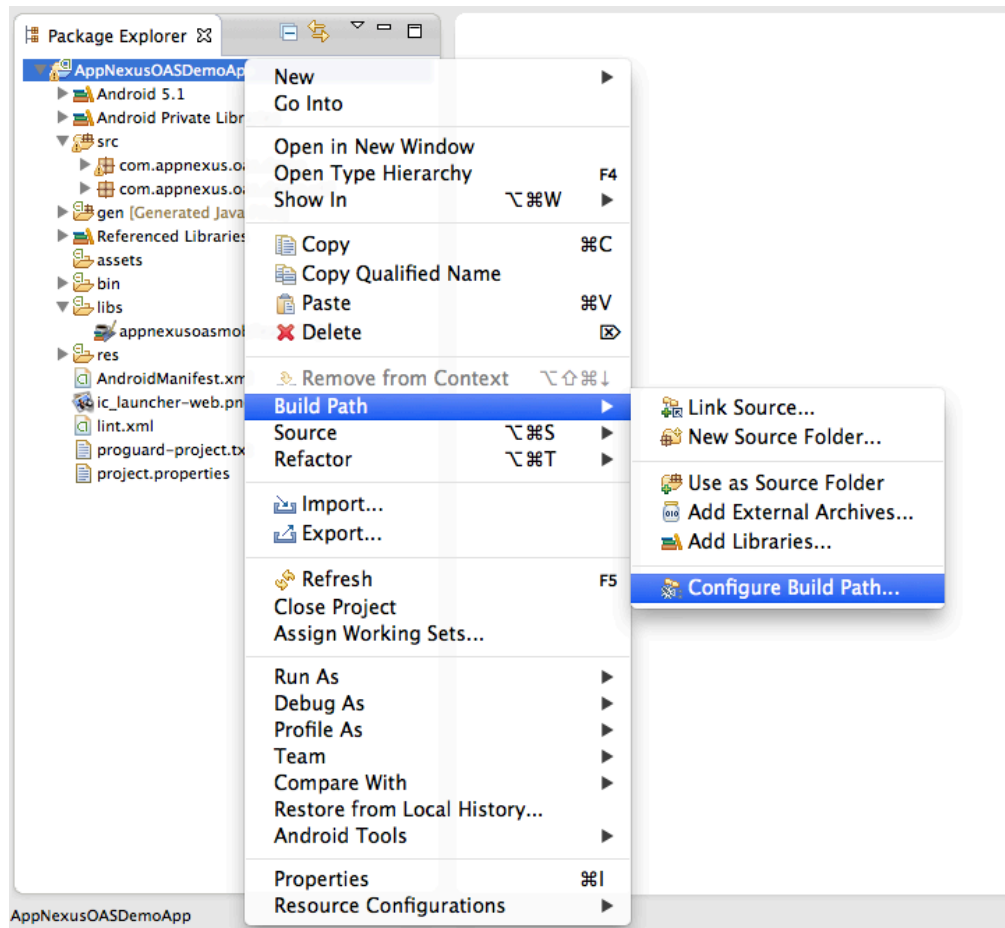




appnexus

28 west 23rd street, floor 4, new york, ny 10010
www.appnexus.com

- Right-click on the project and select Build → Configure Build Path menu option.

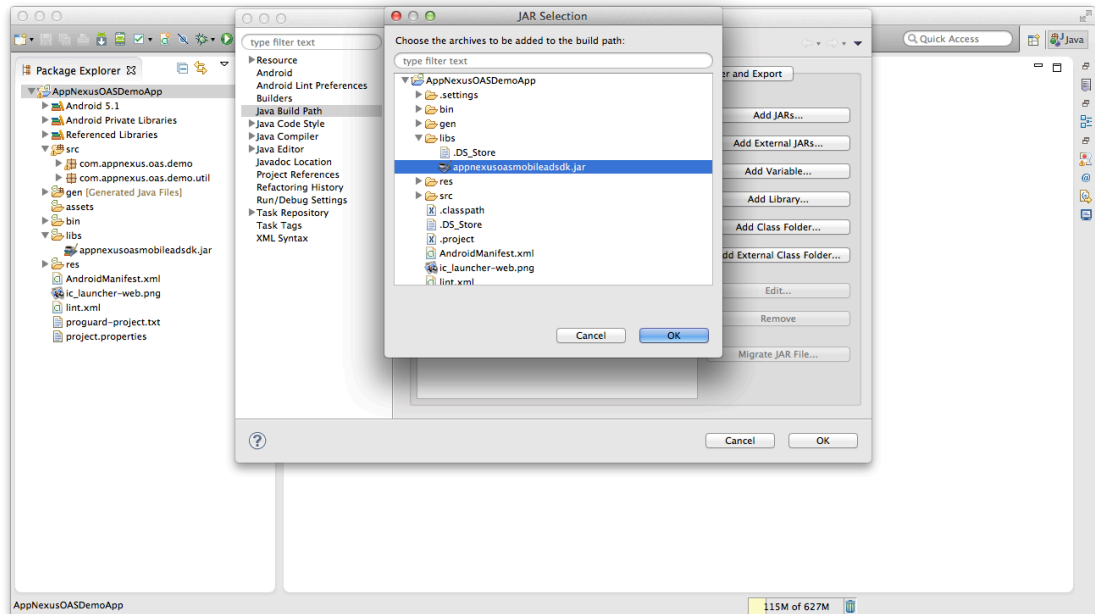




appnexus

28 west 23rd street, floor 4, new york, ny 10010
www.appnexus.com

- Click “Add JARS...” button and add appnexusoasmobileadsdk.jar from your project’s libs directory.

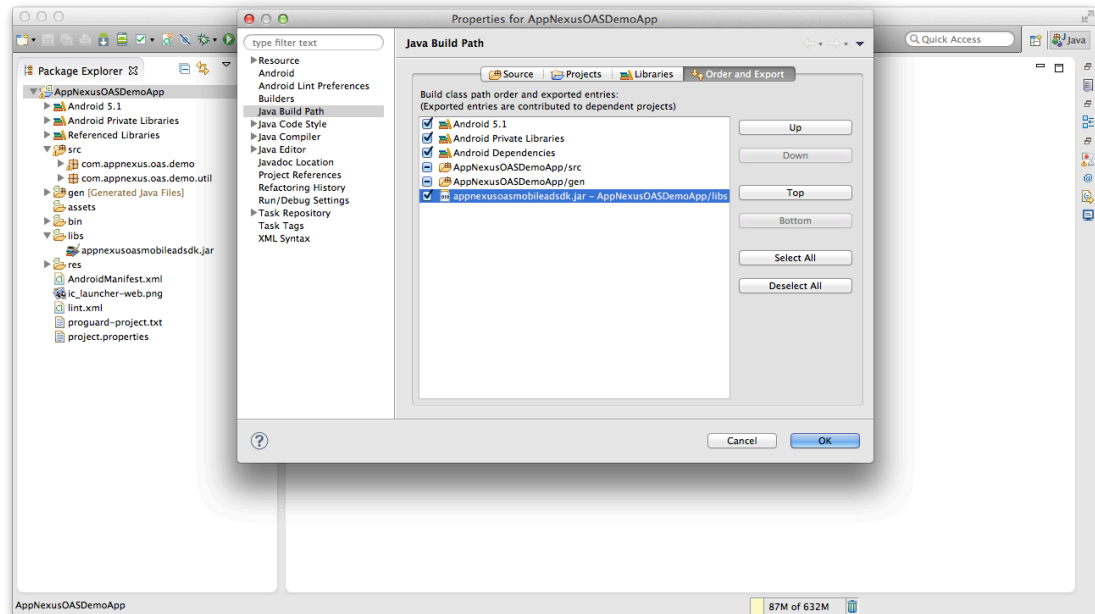




appnexus

28 west 23rd street, floor 4, new york, ny 10010
www.appnexus.com

- Select Order and Export tab and check the appnexusoasmobileadsdk.jar, then press Ok.



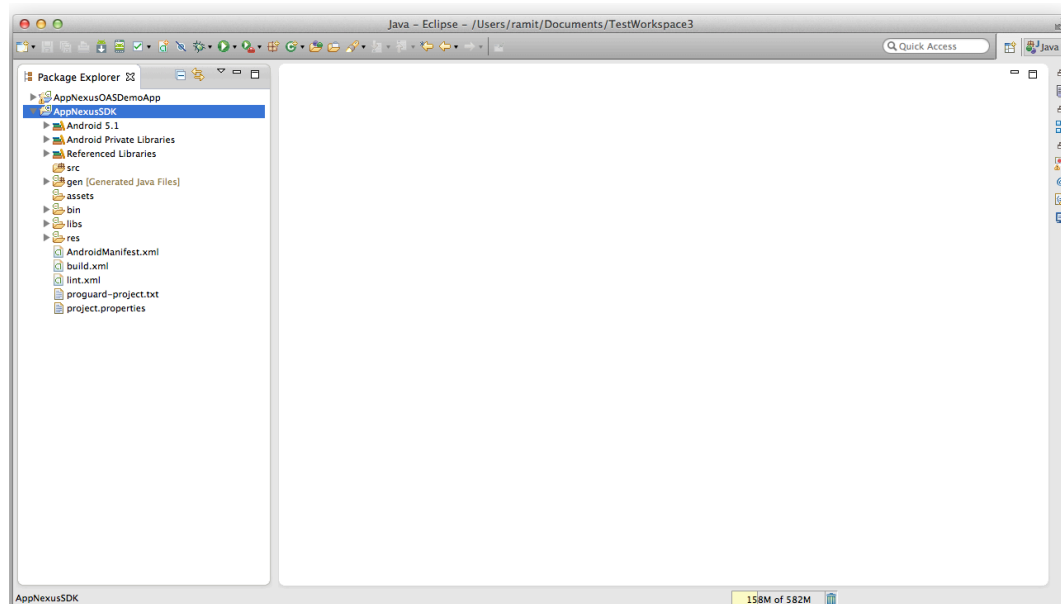
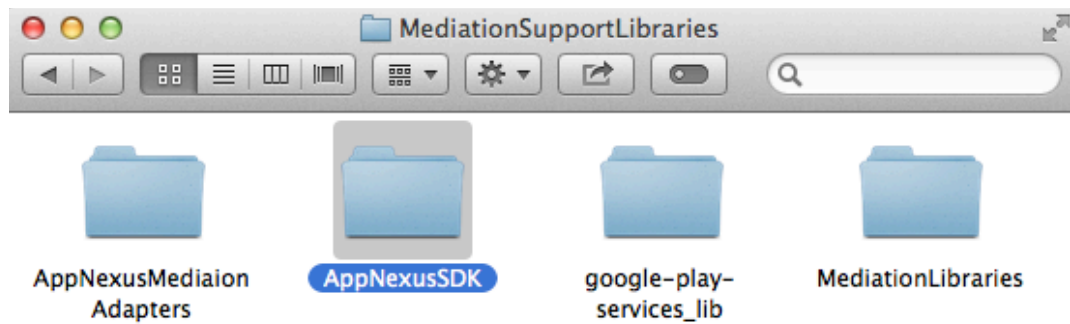


appnexus

28 west 23rd street, floor 4, new york, ny 10010
www.appnexus.com

- **Integrate AppNexusSDK library.**

- Locate the AppNexusSDK library project from the extracted Android_SDK folder.
- Import the library project into your Eclipse workspace. Click File > Import, select Android > Existing Android Code into Workspace, and browse to the copy of the AppNexusSDK library project to import it.

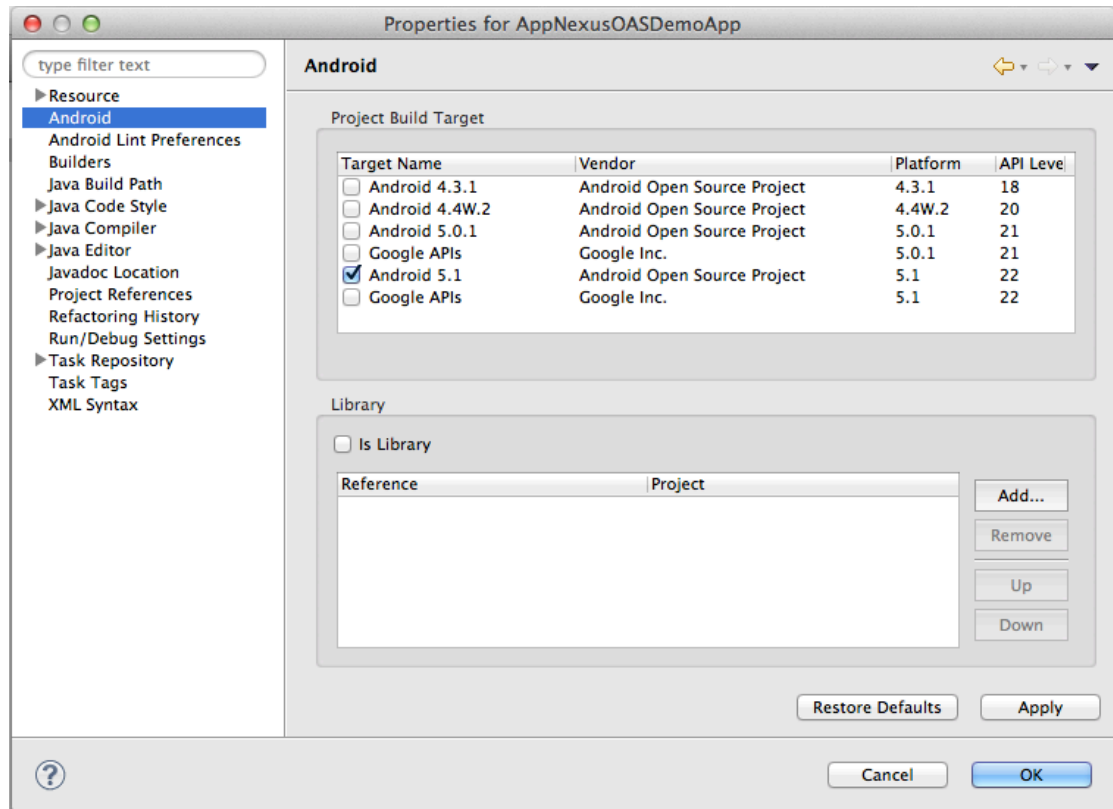




appnexus

28 west 23rd street, floor 4, new york, ny 10010
www.appnexus.com

- Set the dependency in your project for AppNexus SDK. Please follow these steps to set dependency:
 - Right click on your project and select properties.
 - In the Properties window, select the Android properties group in the left pane and locate the Library properties in the right pane.
 - Click Add to open the Project Selection dialog.

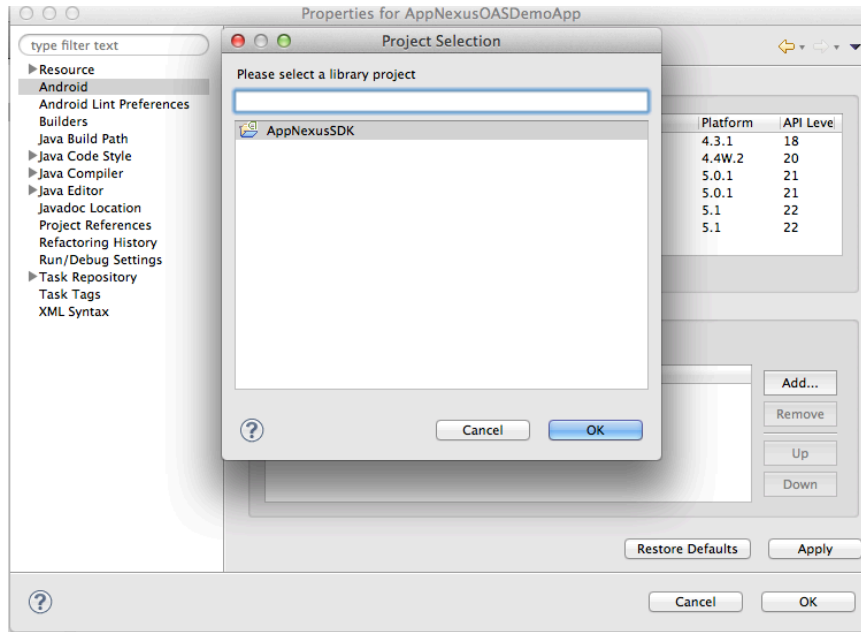




appnexus

28 west 23rd street, floor 4, new york, ny 10010
www.appnexus.com

- From the list of available library projects, select the AppNexusSDK and click OK.



- Then press OK again in the properties window.

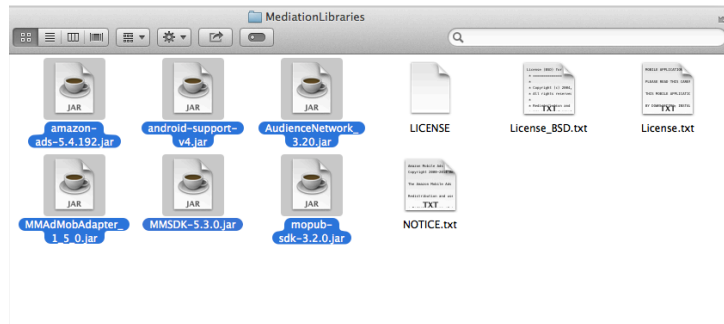




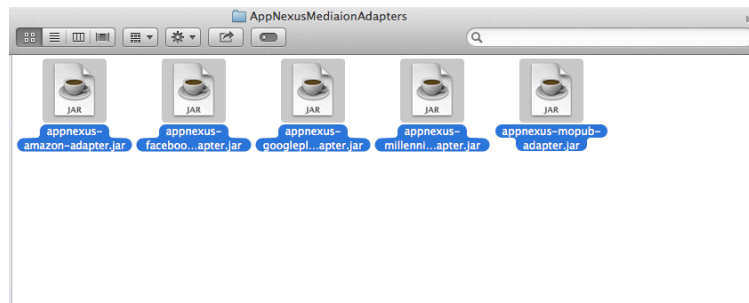
appnexus

28 west 23rd street, floor 4, new york, ny 10010
www.appnexus.com

- **Integrate Mediated libraries and corresponding adapters.**
 - Copy all the libraries present in the MediatedLibraries folder and paste it in the libs folder of your project.



- Similarly copy all the adapter jars from AppNexusMediationAdapters folder and paste it in the libs folder of your project.

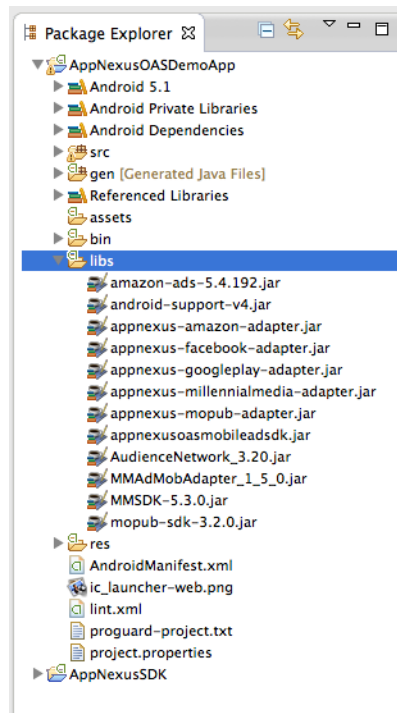




appnexus

28 west 23rd street, floor 4, new york, ny 10010
www.appnexus.com

- libs folder of the application project should look like this:

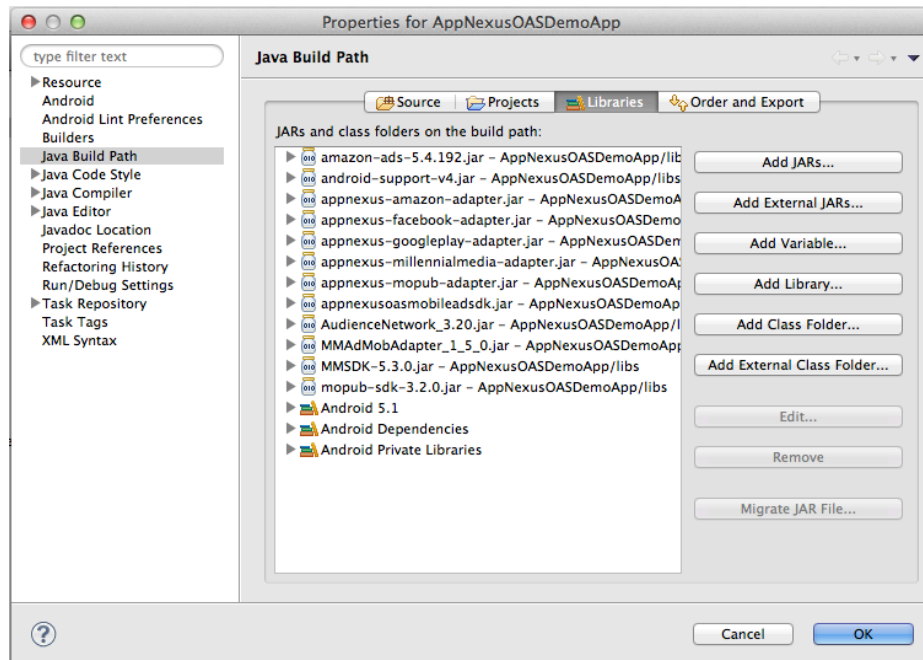




appnexus

28 west 23rd street, floor 4, new york, ny 10010
www.appnexus.com

- Right-click on the project and select Build → Configure Build Path menu option.
- Click Add Jars button and add all the jars present in libs folder of the application.

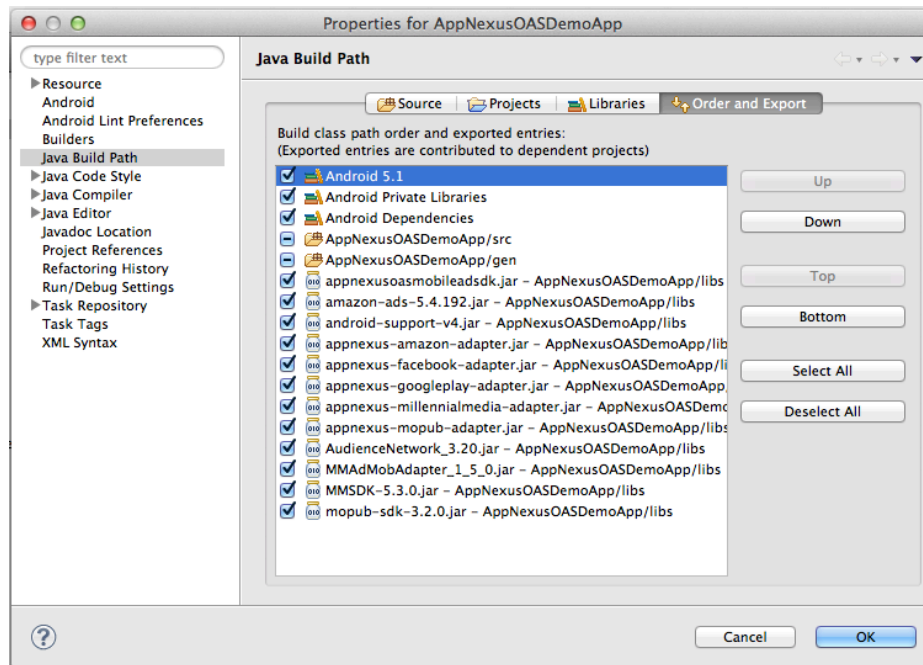




appnexus

28 west 23rd street, floor 4, new york, ny 10010
www.appnexus.com

- Select Order and Export tab and check all the libraries.



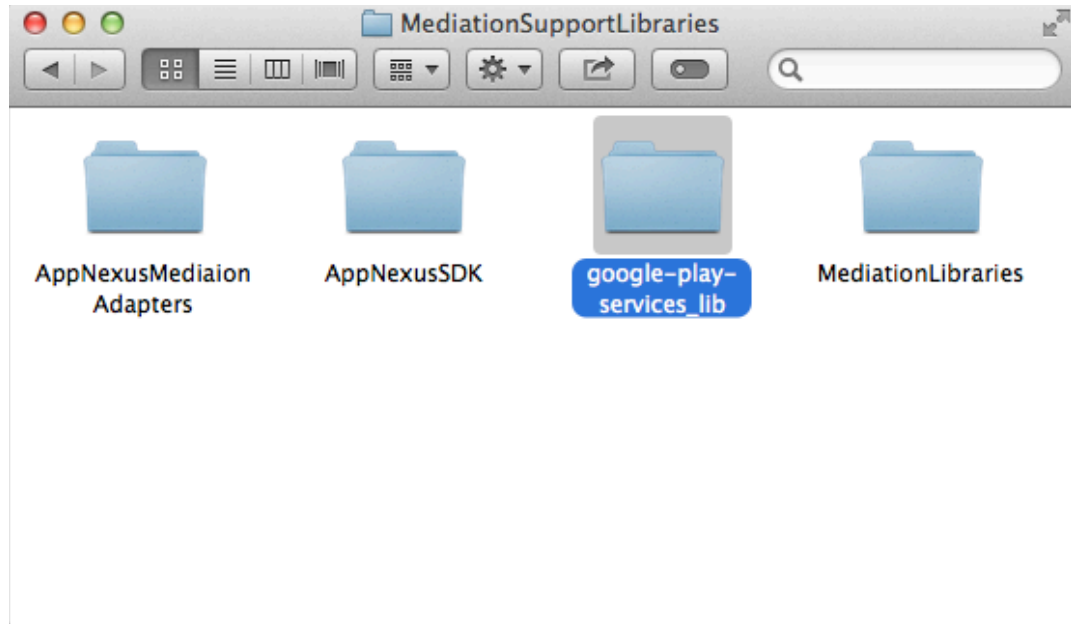
- Press OK and clean build your project.



appnexus

28 west 23rd street, floor 4, new york, ny 10010
www.appnexus.com

- **Integrate Google Play service library.**
 - Import the google-play-services_lib into eclipse from the extracted Android_SDK folder

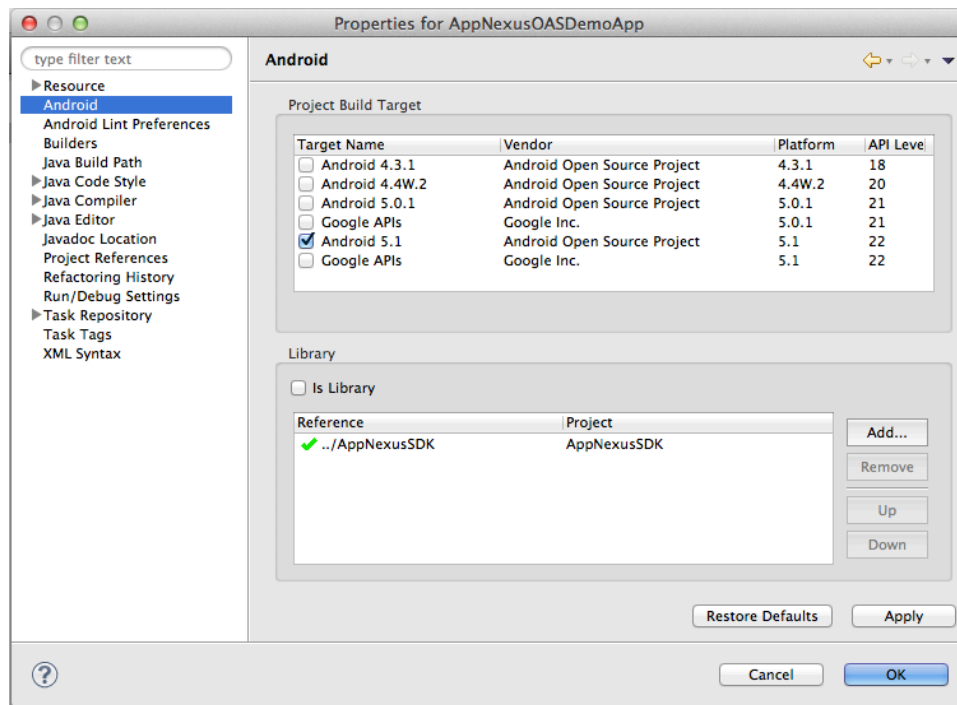




appnexus

28 west 23rd street, floor 4, new york, ny 10010
www.appnexus.com

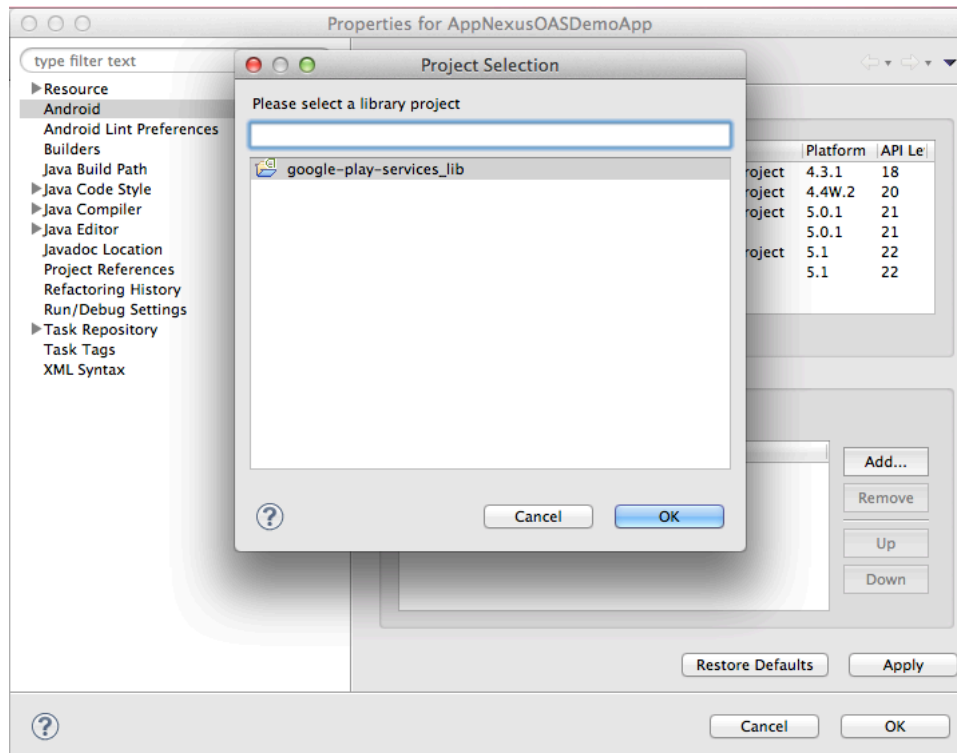
- Set the dependency in your project. Please follow these steps to set dependency:
 - Right click on your project and select properties.
 - In the properties window, press Add... button in the Library section.
 - Select the google-play-services_lib from the list of other options and press OK.
 - Then press OK again in the properties window.
 - Please refer the screenshots below.





appnexus

28 west 23rd street, floor 4, new york, ny 10010
www.appnexus.com



- Alternatively, google-play-services_lib can also be imported from adt bundle folder. Here are the steps to import the same as provide by android user guide.
 - Copy the library project at `<android-sdk>/extras/google/google_play_services/libproject/google-play-services_lib/` to the location where you maintain your Android app projects.
 - Import the library project into your Eclipse workspace. Click File > Import, select Android > Existing Android Code into Workspace, and browse to the copy of the library project to import it.
 - Make sure that latest Google Play Services is installed from the SDK manager.





appnexus

28 west 23rd street, floor 4, new york, ny 10010
www.appnexus.com

Enabling mediation

With OAS Mobile SDK, mediation can be enabled at the application level or at individual slot levels.

When mediation is enabled at application level, it will be enabled throughout the application. Meaning, if any of the banners or interstitial (except Pre-roll) fails to receive ad from OAS, it would try to fill the ad slot from AppNexus, or through mediated ad networks. But it requires publishers to set the mediation placement id along width and height parameters (for mediated banner ads) in slot configuration.

On the other hand, when mediation is enabled at slot levels, the mediation is enabled only for that slot. Meaning, if mediation is enabled for a banner ad on screen 3, then OAS Mobile SDK would try and fetch ads from mediation only for that particular banner placement on screen 3 and not for any other placements through out the application. This way, OAS Mobile SDK provides flexibility to the publishers to enable or disable mediation at individual slot levels as well as at application level.

Enabling Mediation at a slot level:

For Banners

1. Enable mediation at slot level by calling the following method on XAdView's configuration:

```
xAdView.getAdSlotConfiguration().setMediationEnabled(true);
```

2. Set mediation placement id required by the AppNexusSDK to initialize the banner ad view.

```
xAdView.getAdSlotConfiguration().setMediationPlacementId("12345");
```

3. Set banner size required by the AppNexusSDK to initialize banner ad view.

```
xAdView.getAdSlotConfiguration().setMediatedBannerSize(320, 50);
```

Alternatively, if initializing through layout xml, following parameters are required to be set to enable mediation:





appnexus

28 west 23rd street, floor 4, new york, ny 10010
www.appnexus.com

```
oas:mediationEnabled="true"  
oas:mediationPlacementId="12345"  
oas:mediationBannerWidth="320"  
oas:mediationBannerHeight="50"
```

For Interstitials

1. Please enable mediation at slot level by calling following method on XInterstitialAdDialog's configuration:

```
xInterstitialAdDialog.getAdSlotConfiguration().setMediationEnabled(true);
```

2. Set mediation placement id required by the AppNexusSDK to initialize the interstitial ad view.

```
xInterstitialAdDialog.getAdSlotConfiguration().setMediationPlacementId("12345");
```

Enabling Mediation at the application level:

To enable mediation at the application level, following method needs to be called at XGlobalConfiguration along with mediation placement id and banner width and height at slot level:

```
XGlobalConfiguration.getInstance().setGlobalMediationEnabled(true);
```

Please Note:

A slot level setting for mediation takes precedence over application level setting. This means that even if mediation is disabled at the application level, but enabled for a particular slot, then upon ad request failure from OAS server, SDK would still request the mediated network to fill that slot.

In case if mediation is enabled at the application level, but disabled for a particular slot, then upon a failed request from OAS server, SDK would terminate the flow and not make any further request to mediated ad networks for this slot.





appnexus

28 west 23rd street, floor 4, new york, ny 10010
www.appnexus.com

Manifest changes

Please add following declarations in the application's manifest file:

For Google Play services, please add the version information in the application tag:

```
<!-- Google Play SDK -->
<meta-data
  android:name="com.google.android.gms.version"
  android:value="@integer/google_play_services_version" />

<activity
  android:name="com.google.android.gms.ads.AdActivity"
  android:configChanges="keyboard|keyboardHidden|orientation|screenSize|screenLayout|uiMode|smallest
  ScreenSize"
  android:label="@string/app_name"
  android:theme="@android:style/Theme.Black.NoTitleBar" >
</activity>
```

For AppNexus SDK, please add following declaration:

```
<!-- AppNexus SDK -->
<activity
  android:name="com.appnexus.opensdk.AdActivity"
  android:configChanges="orientation|keyboardHidden|screenLayout|uiMode|screenSize"
  android:theme="@android:style/Theme.Black.NoTitleBar" >
</activity>
```

For Mopub SDK, please add following declarations:

```
<!-- MoPub SDK -->
<activity
  android:name="com.mopub.mobileads.MoPubActivity"
  android:configChanges="orientation|keyboardHidden|screenLayout|uiMode|screenSize"
  android:theme="@android:style/Theme.Black.NoTitleBar" >
</activity>

<activity
  android:name="com.mopub.mobileads.MraidVideoPlayerActivity"
```





appnexus

28 west 23rd street, floor 4, new york, ny 10010
www.appnexus.com

```
android:configChanges="orientation|screenSize"
android:label="@string/app_name"
android:theme="@android:style/Theme.Black.NoTitleBar" />
<activity
  android:name="com.mopub.mobileads.MraidActivity"
  android:configChanges="keyboardHidden|orientation|screenSize"
  android:label="@string/app_name"
  android:theme="@android:style/Theme.Black.NoTitleBar" >
</activity>
<activity
  android:name="com.mopub.common.MoPubBrowser"
  android:configChanges="keyboardHidden|orientation|screenSize"
  android:label="@string/app_name"
  android:theme="@android:style/Theme.Black.NoTitleBar" >
</activity>
```

For Millennial Media SDK, please add following declarations:

```
<!-- Millennial Media SDK -->
<activity
  android:name="com.millennialmedia.android.MMAActivity"
  android:configChanges="keyboardHidden|orientation|keyboard"
  android:theme="@android:style/Theme.Translucent.NoTitleBar" >
</activity>
```

For Amazon SDK, please add following declarations:

```
<!-- Amazon SDK -->
<activity
  android:name="com.amazon.device.ads.AdActivity"
  android:configChanges="keyboardHidden|orientation|screenSize">
</activity>
```

Lastly for OAS Mobile SDK, please add following declarations:

```
<!-- AppNexusOASSDK -->
<activity
  android:name="com.appnexus.oas.mobilesdk.ui.custom.XInAppBrowser"
  android:configChanges="orientation|keyboardHidden|screenLayout|uiMode|screenSize" >
```





appnexus

28 west 23rd street, floor 4, new york, ny 10010
www.appnexus.com

</activity>

Setting Targeting parameters for Mediated Banners (Optional)

Targeting parameters can be set on the ad slot configuration by using following method:

To set age as a targeting parameter for mediation, call the following method in xAdView's ad slot configuration:

```
xAdView.getAdSlotConfiguration().setAgeForMediation ("25");
```

To set gender as a targeting parameter for mediation, call the following method in xAdView's ad slot configuration:

```
xAdView.getAdSlotConfiguration().setGenderForMediation(MEDIATION_GENDER.MALE);
```

To set custom targeting parameters for mediation, call the following method on xAdView's ad slot configuration:

```
xAdView.getAdSlotConfiguration().addCustomKeywordsForMediation("key", "value");
```

Setting Targeting parameters for Mediated Interstitials (Optional)

Targeting parameters can be set on the ad slot configuration by using the following method:

To set age as a targeting parameter for mediation, call following method in XInterstitialDialog's ad slot configuration:

```
interstitialDialog.getAdSlotConfiguration().setAgeForMediation ("25");
```





appnexus

28 west 23rd street, floor 4, new york, ny 10010
www.appnexus.com

To set gender as a targeting parameter for mediation, call the following method in XInterstitialDialog's ad slot configuration:

```
interstitialDialog.getAdSlotConfiguration().setGenderForMediation(MEDIATION_GENDER.MALE);
```

To set custom targeting parameters for mediation, call the following method on XInterstitialDialog's ad slot configuration:

```
interstitialDialog.getAdSlotConfiguration().addCustomKeywordsForMediation("key", "value");
```

Setting Location Targeting parameters for Mediated Ads (Optional)

To set location targeting parameters for mediation, call the following method on XGlobalConfiguration:

```
XGlobalConfiguration.getInstance().setMediationTargetLocation(location);
```

This location object can be obtained by using the following code snippet in the application:

```
import android.location.LocationManager;  
import android.location.Location;
```

```
// Get the current location, and add it to the XGlobalConfiguration.
```

```
LocationManager locationManager = (LocationManager)  
getApplicationContext().getSystemService(LOCATION_SERVICE);
```

```
Location location =  
locationManager.getLastKnownLocation(LocationManager.NETWORK_PROVIDER);  
XGlobalConfiguration.getInstance().setMediationTargetLocation(location);
```





appnexus

28 west 23rd street, floor 4, new york, ny 10010
www.appnexus.com

Enabling Mediation Dynamically

To enable mediation dynamically through OAS, publishers can use the following XML format to traffic in OAS as the house ad. Upon receiving this response, SDK (version 2.1.0 and above) will treat it as a no ad response and enable mediation behind the scenes with the updated placement id and other parameters as mentioned in the xml. If app developer has already specified a placement id in SDK's configuration, then placement id specified in the XML will take precedence.

This XML needs to be trafficked in OAS with the lowest priority and no frequency capping or limits, so OAS would serve this XML only when there are no other ads available.

```
<?xml version="1.0" encoding="UTF-8"?>
<MobileSDK>
  <ClientSideMediation placementId="12345" width="320" height="50" />
</MobileSDK>
```

Note:

- AdType attribute for the above XML must be set as text/xml for SDK to detect correctly.
- "width" and "height" attributes are optional, as it is required only by the banner ads. For interstitial ads width and height is diligently ignored. Also, since XML is case sensitive, it is important to maintain the letter cases as it is. Only the values for placementId, width and height are allowed to vary.

Mediation Libraries with Adapters

OAS SDK supports 5 different types of mediated ad networks. Following is the list of supported ad networks with their library and corresponding AppNexus adapter.

1. Amazon

- Library name: amazon-ads-5.4.192.jar





appnexus

28 west 23rd street, floor 4, new york, ny 10010
www.appnexus.com

- Adapter name: appnexus-amazon-adapter.jar

2. Facebook

- Library name: AudienceNetwork_3.20.jar
- Adapter name: appnexus-facebook-adapter.jar

3. Mopub

- Library name: mopub-sdk-3.2.0.jar
- Adapter name: appnexus-mopub-adapter.jar

4. AdMob/Google Play/DFP

- Library name: google-play-services_lib library project
- Adapter name: appnexus-googleplay-adapter.jar

5. Millennial Media

- Library name: MMSDK-5.3.0.jar
- Adapter name: appnexus-millennialmedia-adapter.jar

Please Note: Millennial Media SDK (MMSDK-5.3.0.jar) can be downloaded separately after setting up account with them.

Debugging Mediation

To debug ad fetch through mediation, OAS Mobile SDK displays the following logs to help application developers debug and troubleshoot some basic integration errors.

Log Message:

Mediation is currently not supported for Pre-roll ads.

Tried to fill the pre-roll slot with ad via mediation, however, mediation is currently not supported for pre-roll ad.

Log Message:





appnexus

28 west 23rd street, floor 4, new york, ny 10010
www.appnexus.com

Cannot mediate. Please enable mediation at slot level or application level.

To fill the slot with ad via mediation, you need to enable mediation at the application level or at the slot level.

Log Message:

Cannot Mediate. Please provide placement id to start receiving mediated ads.

To fill slots via mediation, placementId needs to be set at the slot configuration.

Log Message:

Cannot Mediate. Please provide mediated banner width and height to start receiving mediated ads.

To fill banner slots via mediation, mediatedBannerWidth and mediatedBannerHeight needs to be set at the slot configuration.

Log Message:

Missing Mediation Classes. AppNexus library is not integrated. Please integrate AppNexus library to start receiving mediated ads.

Publisher has not integrated the mediation library. To fill the slot with mediated ads, the AppNexus library must be integrated.

