

Advertising SDK for Cocos2dx

1. Android plugin integration

The Advertising SDK allows developers to support a lot of ads types including banner, interstitial, video, landing page and native ads.

1.1. General requirements

Android SDK 4.0.3 (API level 15) or higher.

Google Play Services to enable Advertising ID support (recommended); installation instructions [here](#)

You may get acquainted with native Android SDK [here](#).

Also you need to install [PwCore module](#) and inherit from AppCompatActivity.

1.2. Manifest changes:

Update your AndroidManifest.xml, include the following permissions and activity:

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

<!-- Inside of the application tag: -->
<activity
    android:name="com.phunware.advertising.internal.PwAdActivity"
    android:configChanges="keyboard|keyboardHidden|orientation|screenSize" />
```

The path to AndroidManifest.xml:

[\\$ExampleProjectPath/Project/projects/proj.android-studio/app/](#)

1.3. Add libraries:

The following libraries is required MaaSAdvertising.aar (you may find it by path [\\$ExampleProjectPath/external/phunware/libs/android](#)). For example you may see instructions [here](#).

Also you need add PWCore-3.0.2.aar. For it you need to add into repository main gradle:

```
maven {  
    url "https://nexus.phunware.com/content/groups/public/"  
}
```

And then compile aar file:

```
compile 'com.phunware.core:core:3.0.2'
```

1.3. Edit strings.xml:

Edit strings.xml in your android studio project. Add the following string:

```
<string name="interstitial">Interstitial</string>  
<string name="video">Video</string>  
<string name="banner">Banner</string>  
<string name="nativeAd">Native</string>  
<string name="native_ads">Native Ads</string>  
<string name="ads">Ads</string>  
<string name="nativeAdNewsFeed">Native Ad</string>  
<string name="app_appid">appid</string>  
<string name="app_accesskey">abcd12345</string>  
<string name="app_signaturekey">098765fghjkl</string>  
<string name="app_encryptionkey">65as6d54f065sd4f0</string>  
<string name="banner_zone_id">7979</string>  
<string name="video_zone_id">75395</string>  
<string name="inrs_zone_id">7983</string>  
<string name="native_zone_id">63289</string>  
<string name="landing_page_zone_id">76663</string>
```

The path to string.xml:

```
$ExampleProjectPath/Project/projects/proj.android-studio/app/res/values
```

1.4. Edit proguard file:

Edit proguard, add following strings:

```
-dontwarn com.google.android.gms.internal.**  
-keep class com.google.android.gms.internal.** {*;}  
-dontwarn com.google.gson.**  
-keep class com.google.gson.** {*;}  
-dontwarn android.support.customtabs.**  
-keep class android.support.customtabs.** {*;}  
-dontwarn com.millennialmedia.android.**
```

```

-keep class com.millennialmedia.android.** {*;}
-dontwarn org.apache.http.**
-keep class org.apache.http.** {*;}
-dontwarn com.squareup.**
-keep class com.squareup.** {*;}
-keep class org.cocos2dx.lib.** {*;}
-keep class com.softeq.** {*;}
-dontwarn com.squareup.**
-keep class com.squareup.** {*;}
-dontwarn com.phunware.**
-keep class com.phunware.** {*;}
-dontwarn retrofit.**
-keep class retrofit.** {*;}

-keep class * implements android.os.Parcelable {
    public static final android.os.Parcelable$Creator *;}

-keepattributes *Annotation*
-dontoptimize
-dontobfuscate

```

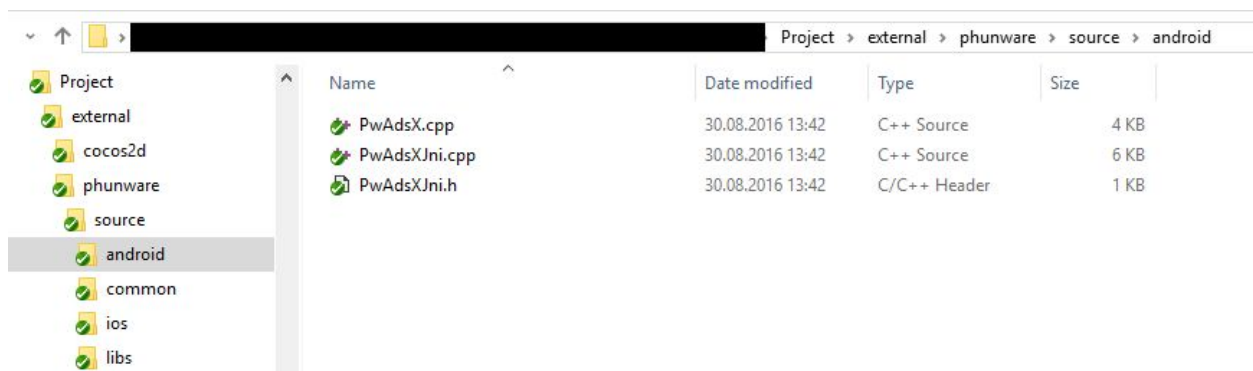
The path to string.xml:

[\\$ExampleProjectPath/Project/projects/proj.android-studio/app/res/values](#)

1.5. Add files to Android.mk:

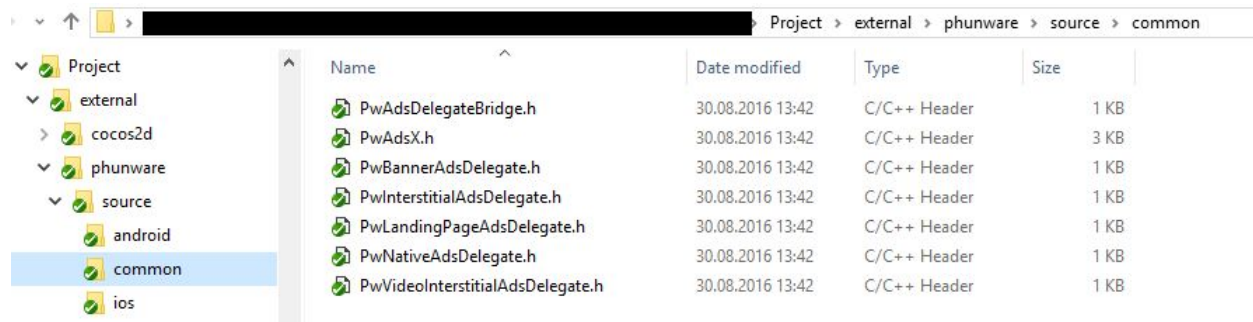
Also you need to add a few files to the project. You may find them in folders:

[\\$ExampleProjectPath/external/phunware/source/android](#)



Screenshot 1. Android source files list

[\\$ExampleProjectPath/external/phunware/source/common](#)



Screenshot 1. Common source files list

Edit your Android.mk:

The path to string.xml:

[\\$ExampleProjectPath/Project/projects/proj.android-studio/app/jni](#)

To imports add:

[\\$\(call import-add-path,\\$\(LOCAL_PATH\)/../../../../external/phunware/source/common\)](#)

Add source files:

```
LOCAL_SRC_FILES := ***\
../../../../external/phunware/source/android/PwAdsX.cpp \
../../../../external/phunware/source/android/PwAdsXJni.cpp
```

And header files:

```
LOCAL_C_INCLUDES := $(LOCAL_PATH)/../../../../source \
$(LOCAL_PATH)/../../../../external/phunware/source/common \
$(LOCAL_PATH)/../../../../external/phunware/source/android
```

The list of requiered source files:

```
PwAdsDelegateBridge.h
PwBannerAdsDelegate.h
PwInterstitialAdsDelegate.h
PwLandingPageAdsDelegate.h
PwRewardedVideoAdsDelegate.h
PwNativeAdsDelegate.h
PwVideoInterstitialAdsDelegate.h
PwAdsX.h
```

PwAdsXJni.h
PwAdsXJni.cpp
PwAdsX.cpp

After all this steps you can open android project and build it, or you may run shell command from folder: `$ExampleProjectPath/Project/projects`

Command:

`cocos compile -p android --android-studio`

2. Ads usage

The Main ads class is PwAdsX. It is singleton class. To get instance of this class you should call static method `PwAdsX::sharedPwAdsX()`.

2.1. Banner Ads Integration

To show banner ads in an application, the following part of the code needs to be added:

```
#include "PwAdsX.h"
...
{
    PwAdsX::sharedPwAdsX()->showBannerAd();
}
```

And to hide banner ads you need to call:

```
#include "PwAdsX.h"
...
{
    PwAdsX::sharedPwAdsX()->hideBannerAd();
}
```

If you want to receive banner ads callbacks from plugin, you need to inherit from `PwBannerAdsDelegate`, and inherited object should be transferred to the plugin.

```
#include "PwBannerAdsDelegate.h"
...
{
    PwAdsX::sharedPwAdsX()->setBannerDelegate(this);
}
```

2.2. Native Ads Integration

To show native ads in an application, the following part of the code needs to be added:

```
#include "PwAdsX.h"
...
{
    PwAdsX::sharedPwAdsX()->showNativeAd();
}
```

And to hide native ads you need to call:

```
#include "PwAdsX.h"
...
{
    PwAdsX::sharedPwAdsX()->hideNativeAd();
}
```

If you want to receive native ads callbacks from plugin you need to inherit from PwNativeAdsDelegate, and inherited object should be transferred to the plugin.

```
#include "PwNativeAdsDelegate.h"
...
{
    PwAdsX::sharedPwAdsX()->setNativeDelegate(this);
}
```

2.3. Video Interstitial Ads Integration

To show video interstitial ads in an application, the following part of the code needs to be added:

```
#include "PwAdsX.h"
...
{
    PwAdsX::sharedPwAdsX()->showVideoInterstitialAd();
}
```

If you want to receive video interstitial ads callbacks from plugin you need to inherit from PwVideoAdsDelegate, and inherited object should be transferred to the plugin.

```
#include "PwVideoAdsDelegate.h"
...
```

```
{  
    PwAdsX::sharedPwAdsX()->setVideoDelegate(this);  
}
```

2.4. Rewarded Video Ads Integration

To show rewarded video ads in an application you need at first load it:

```
#include "PwAdsX.h"  
...  
{  
    PwAdsX::sharedPwAdsX()->loadRewardedVideoAd(isTestMode);  
}
```

And then the following part of the code needs to be added to show rewarded video ad:

```
#include "PwAdsX.h"  
...  
{  
    PwAdsX::sharedPwAdsX()->showRewardedVideoAd();  
}
```

If you want to receive rewarded video callbacks from plugin you need to inherit from PwRewardedVideoAdsDelegate, and inherited object should be transferred to the plugin.

```
#include "PwVideoAdsDelegate.h"  
...  
{  
    PwAdsX::sharedPwAdsX()->getRewardedVideoDelegate(this);  
}
```

2.5. Interstitial Ads Integration

To show interstitial ads in an application, the following part of the code needs to be added:

```
#include "PwAdsX.h"  
...  
{  
    PwAdsX::sharedPwAdsX()->showInterstitialAd();  
}
```

If you want to receive interstitial ads callbacks from plugin you need to inherit a class from PwInterstitialAdsDelegate, and inherited object should be transferred to the plugin.

```
#include "PwInterstitialAdsDelegate.h"
...
{
    PwAdsX::sharedPwAdsX()->setInterstitialDelegate(this);
}
```

2.6. Landing Page Ads Integration

To show landing page ads in an application, the following part of the code needs to be added:

```
#include "PwAdsX.h"
...
{
    PwAdsX::sharedPwAdsX()->showLandingPageAd();
}
```

If you want to receive landing page ads callbacks from plugin you need to inherit a class from PwLandingPageAdsDelegate, and inherited object should be transferred to the plugin.

```
#include "PwLandingPageAdsDelegate.h"
...
{
    PwAdsX::sharedPwAdsX()->setLandingPageDelegate(this);
}
```