

# Unity LevelPlay (formerly ironSource) Adapter Installation Guide

Android LevelPlay Adapter Installation

## Part 1: Adding AdView as a Custom Network in LevelPlay Dashboard

### Step 1: Log in to your LevelPlay account

You must have an active LevelPlay account and an app created in it to integrate AdView.

For complete setup instructions, refer to the official guide:

[How to start monetizing your first app - LevelPlay Knowledge Center](#)

### Step 2: Add your App

Log in to your LevelPlay account → Click **New App**.

The screenshot shows the 'Add new app' form in the Unity LevelPlay dashboard. The left sidebar contains navigation links: Home, Apps, Mediation, Reports, Performance, Report generator, Real time pivot, Cohorts, Activity logs, Management, Segments, A/B, Setup, Ad units, Instances, Networks, Placements, and Direct deals. The main form has a title 'Add new app' and a link 'How to integrate the SDK'. It contains three sections: 'Settings' with fields for 'Store availability' (radio buttons for 'Live app' and 'Not live yet'), 'App name' (text input 'MyTest'), 'Platform' (radio buttons for 'iOS' and 'Android'), and 'Setup configuration' (checkbox for 'Duplicate from another app'); 'App permissions' with a 'Restricted users' dropdown; and 'Privacy' with a link to 'ironSource Ads'.

- If your app is already live on Google Play or App Store → Enter the store URL → Click **Import App Info**.
- If not live yet → Use a temporary name (you can update it later).

The screenshot shows the 'Edit app' form in the Unity LevelPlay dashboard. The left sidebar is the same as in the previous screenshot. The main form has a title 'Edit app' and a link 'How to integrate the SDK'. It contains two sections: 'Settings' with fields for 'App key' (2432d4495), 'Bundle ID' (com.adview.easyweather.weather), 'App name' (text input '简单天气'), and 'Category' (Casino: Poker); and 'App permissions' with a 'Restricted users' dropdown.

## Step 3: Configure App Settings

Set COPPA and app permissions as required → Click **Save and set up ad units**.

After saving, you will receive your **APP\_KEY** (required for SDK initialization).

### ← Add new app

App name \* MyTest

Platform \* ☐ iOS ☒ Android

Setup configuration ☐ Duplicate from another app

**App permissions**

Restricted users ? Select

Privacy ironSource Ads

COPPA \* ? Select

Cancel Save Save and set up ad units

## Step 4: Select your ad units

Choose the ad formats you want to mediate:

- Rewarded Video
- Interstitial
- Banner

← Create new ad unit

MyTest

**Ad unit setup**

Ad unit name \* my\_banner 9/255

Ad format \* Banner

Setup configuration \* ☒ New setup ☐ Duplicate setup

Banner refresh rate \* ? 25 sec

Cancel Save

You can add more formats later.

## Step 5: Add AdView Custom Adapter

1. Go to **Setup** → **Networks** menu
2. Click **Add custom network**

**Manage ad networks**

Your account is pending approval. We'll notify you by email when your account is approved.

Displaying 29 out of 29 ad networks

Network	Supported ad formats	
ironSource Ads	[Icons]	
AdView	[Icons]	
Unity Ads	[Icons]	Setup
APPLLOVIN	[Icons]	Setup
amazon publisher services	[Icons]	Setup
+ Add custom network		Setup

3. Enter the Network Key:

Adview Network key:15bed2985

#### ← Ad network setup

**Custom network settings**

Network key \*

4. Click **Save**

## Step 6: Add AdView Custom Network

1. In the **Instances** tab, find **Custom**.
2. Select the AdView network you just created

**Instances**

Your account is pending approval. We'll notify you by email when your account is approved.

Displaying 2 out of 2 ad networks

Bidding (1)	Rewarded	Interstitial	Banner	Native
ironSource	✓	✓	✓	✓
Custom (1)	Rewarded	Interstitial	Banner	Native
AdView	✓	✓	✓	N/A

3. Enter your **APPID** and **Placement ID** (obtained from AdView platform)
4. Click **Save**

Setup complete!

You should now see at least two networks under **Ad units & instances**: ironSource + AdView.

## Part 2: Android Integration – AdView Adapter

### Step 1: Add AdView SDK & Adapter to Your Project

#### Option 1 – Recommended (Maven Central)

In your project-level `build.gradle`:

```
allprojects {
    repositories {
        mavenCentral()
    }
}
```

In your app module `build.gradle`:

```
implementation 'com.adview:android-levelplay-adapter:4.x.x'
```

If you want to use AdView sdk features, such as COPPA, you should add following:

```
implementation 'com.adview:android-sdk:4.x.x'
```

## Option 2 – Manual AAR Inclusion

Place these files in `libs` folder:

- `AdViewSDK_Android-4.x.x.aar`
- `com.adview-android-levelplay-adapter-4.x.x.aar`

Then add:

```
dependencies {  
    api fileTree(include: ['*.aar'], dir: 'libs')  
}
```

Sync Gradle after adding dependencies.

## Step 2: AndroidManifest.xml Configuration Add the following activities inside `<application>` tag:

```
<application  
    <activity  
        android:name="com.advg.video.AdViewVideoActivity"  
        android:configChanges="keyboardHidden|orientation|screenSize"  
        android:hardwareAccelerated="true" >  
    </activity>  
    <activity android:name="com.advg.utils.AdActivity"  
        android:theme="@android:style/Theme.Translucent.NoTitleBar" />  
    <activity android:name="com.advg.utils.AdViewLandingPage"  
        android:configChanges="keyboardHidden|orientation|screenSize"  
    />  
</application>
```

## Step 3: Initialize LevelPlay SDK

```
LevelPlayInitRequest initRequest = new  
LevelPlayInitRequest.Builder(APP_KEY).build()  
LevelPlay.init(this, initRequest, new DemoInitializationListener(this));
```

In `onInitSuccess`:

```
public void onInitSuccess(@NonNull LevelPlayConfiguration configuration) {  
    logCallbackName(TAG, "");  
    this.listener.createInterstitialAd();  
    this.listener.createRewardedAd();  
    this.listener.createBannerAd();  
}
```

**Important:** If using COPPA, initialize AdView SDK first (see COPPA section below).

## Banner Integration

You need set ad unit id and set LevelPlayBannerAdViewListener to handle the ad events.

### create Banner

First , listener LevelPlayBannerAdViewListener like the following and you should implement them.

```
public interface LevelPlayBannerAdViewListener {
    void onAdLoaded(@NotNull LevelPlayAdInfo var1);
    void onAdLoadFailed(@NotNull LevelPlayAdError var1);
    void onAdDisplayed(@NotNull LevelPlayAdInfo var1);
    default void onAdDisplayFailed(@NotNull LevelPlayAdInfo var1, @NotNull LevelPlayAdError var2);
    default void onAdClicked(@NotNull LevelPlayAdInfo var1) {
        Intrinsics.checkNotNullParameter(var1, "adInfo");
    }
    default void onAdExpanded(@NotNull LevelPlayAdInfo var1) {
        Intrinsics.checkNotNullParameter(var1, "adInfo");
    }
}
```

Note: size is LevelPlayAdSize.BANNER, also LARGE, MEDIUM\_RECTANGLE can be used.

```
// 1. recommended - Adaptive ad size that adjusts to the screen width
// LevelPlayAdSize adSize = LevelPlayAdSize.createAdaptiveAdSize(this);
// 2. Adaptive ad size using fixed width ad size
// LevelPlayAdSize adSize = LevelPlayAdSize.createAdaptiveAdSize(this, 400);
// 3. Specific banner size - BANNER, LARGE, MEDIUM_RECTANGLE
LevelPlayAdSize adSize = LevelPlayAdSize.BANNER;
// Create the banner view and set the ad unit id and ad size
LevelPlayBannerAdView bannerAd;
LevelPlayBannerAdView.Config config = new
    LevelPlayBannerAdView.Config.Builder().setAdSize(adSize).build();
bannerAd = new LevelPlayBannerAdView(this, BANNER_AD_UNIT_ID, config);
// set the banner listener
bannerAd.setBannerListener(new DemoBannerAdListener(this));
// add LevelPlayBannerAdView to your container
FrameLayout.LayoutParams layoutParams = new FrameLayout.LayoutParams(
    MATCH_PARENT, MATCH_PARENT);
bannerParentLayout.addView(bannerAd, 0, layoutParams);
```

### load banner

```
bannerAd.loadAd();
```

### destroy banner

```
bannerAd.destroy();
```

## Interstitial

For interstitial ad, need load and show ad 2 steps. at first you need implement the following interfaces of Interstitial ads, the LevelPlayInterstitialAdListener like following:

```
public interface LevelPlayInterstitialAdListener {
    void onAdLoaded(@NotNull LevelPlayAdInfo var1);
    void onAdLoadFailed(@NotNull LevelPlayAdError var1);
    void onAdDisplayed(@NotNull LevelPlayAdInfo var1);
    void onAdDisplayFailed(@NotNull LevelPlayAdError var1, @NotNull
LevelPlayAdInfo var2)
    void onAdClicked(@NotNull LevelPlayAdInfo var1) {
        Intrinsics.checkNotNullParameter(var1, "adInfo");
    }
    void onAdClosed(@NotNull LevelPlayAdInfo var1) {
        Intrinsics.checkNotNullParameter(var1, "adInfo");
    }
    void onAdInfoChanged(@NotNull LevelPlayAdInfo var1) {
        Intrinsics.checkNotNullParameter(var1, "adInfo");
    }
}
```

imp the interfaces, such as following:

```
@Override
public void onAdLoaded(@NonNull LevelPlayAdInfo adInfo) {
    logCallbackName(TAG, "adInfo = " + adInfo);
}
/**
    Called after an interstitial ad has attempted to load but failed
    @param error The reason for the error
 */
@Override
public void onAdLoadFailed(@NonNull LevelPlayAdError error) {
    logCallbackName(TAG, "error = " + error);
}
....
```

and then, you can set listener to it, for example, if you implement listeners in activity, you can use the following:

#### create ad

```
LevelPlayInterstitialAd interstitialAd;
interstitialAd = new LevelPlayInterstitialAd(INTERSTITIAL_AD_UNIT_ID);
interstitialAd.setListener(new DemoInterstitialAdListener(this));
```

#### load ad

```
interstitialAd.loadAd();
```

#### show ad

After ad is loaded. you can show it anytime. (demo codes)

```

if (interstitialAd != null && interstitialAd.isAdReady()) {
    // This will present the Interstitial.
    log("showAd for interstitial");
    interstitialAd.showAd(this);
}

```

## Reward Video

Reward ads also need 2 steps: load and show. first you need implement the following interfaces of Reward video ads.

### create reward ad

```

public interface LevelPlayRewardedAdListener {
    void onAdLoaded(@NotNull LevelPlayAdInfo var1);
    void onAdLoadFailed(@NotNull LevelPlayAdError var1);
    void onAdDisplayed(@NotNull LevelPlayAdInfo var1);
    void onAdRewarded(@NotNull LevelPlayReward var1, @NotNull LevelPlayAdInfo var2);
    default void onAdDisplayFailed(@NotNull LevelPlayAdError var1, @NotNull LevelPlayAdInfo var2) {}
    default void onAdClicked(@NotNull LevelPlayAdInfo var1) {}
    default void onAdClosed(@NotNull LevelPlayAdInfo var1) {}
    default void onAdInfoChanged(@NotNull LevelPlayAdInfo var1) {
        Intrinsics.checkNotNullParameter(var1, "adInfo");
    }
}

```

and then, you can set listener to it, for example , if you implement listeners in activity, you can use the following:

```

LevelPlayRewardedAd rewardedAd = new
LevelPlayRewardedAd(REWARDED_VIDEO_AD_UNIT_ID);
rewardedAd.setListener(new DemoRewardedVideoAdListener(this));

```

### load reward

```

if (rewardedAd != null) {
    log("loadAd for rewarded ad");
    rewardedAd.loadAd();
}

```

### show reward

After ad is loaded. you can show it anytime. (demo codes)

```

if (rewardedAd != null && rewardedAd.isAdReady()) {
    // This will present the Rewarded Video.
    log("show ad for rewarded ad");
    rewardedAd.showAd(this);
}

```

For full callback implementations and detailed integration, refer to official docs:



- [Banner Integration](#)
- [Interstitial Integration](#)
- [Rewarded Video Integration](#)

## COPPA flagging

In the application of the COPPA regulations, to indicate that the content of your app is intended for children under 13 years of age.

LevelPlay not has such interface from adapter, so you need set the flag with AdView sdk.

initialize sdk.

```
AdViewSdk.initialize(Context ctx);
```

and then use this:

```
void setCoppa(boolean coppa);
```

example codes:

```
//init adview sdk before levelplay initialize
AdViewSdk.initialize(this);
AdViewSdk.setCoppa(true); //set coppa
...
//levelplay init
LevelPlay.init(this, initRequest, new DemoInitializationListener(this));
```

The value declared (true or false) in the argument of the **setCoppa** function will determine whether advertising restrictions will be activated or not.

You must set the child-specific content tag before LevelPlay initialize.

## ProGuard setting

You should add the following rules in proguard-project.txt , or you may not run sdk pass through .

```
-keep public class android.webkit.JavascriptInterface { *; }
-dontwarn com.iab.omid.library.adview.**
-keep public class com.iab.omid.library.adview.**.* { *; }

-dontwarn com.advg.**
-keep public class com.advg.**.* { *; }
```

and also, adview's adapter codes also not be obfuscated, adview 's custom adapter class's package path is :

```
package com.ironsource.adapters.custom.adview.AdViewCustomAdapter;
```

so you must declare the following definition in proguard-project.txt :

```
-keep public class com.ironsource.adapters.custom.adview { *; }
```

## **You're all done!**

Your LevelPlay SDK should start showing AdView ads immediately.

Otherwise please contact your AM or [partner@adview.com](mailto:partner@adview.com).