



UnityPlugin for Android User Guide

개요

애드립 Unity 플러그인 적용을 위한 문서입니다.

애드립 Unity 플러그인을 사용하기 위해서 다음이 필요합니다.

- 애드립 API 키
- 애드립 최신 SDK
- 각 광고 플랫폼 사이트에서 발급받은 APP - ID 및 최신 SDK

애드립 SDK 다운부터 API 키 발급까지에 대한 과정은 아래 링크에서 확인하실 수 있습니다.

<http://adlibr.com/howto/prepare>

이 애드립 Unity 플러그인은 Unity 4.3.3 에서 작성 되었으며, 이전 버전에 대해서는 테스트가 이루어 지지 않았습니다.

오류 보고 또는 추가 문의사항은

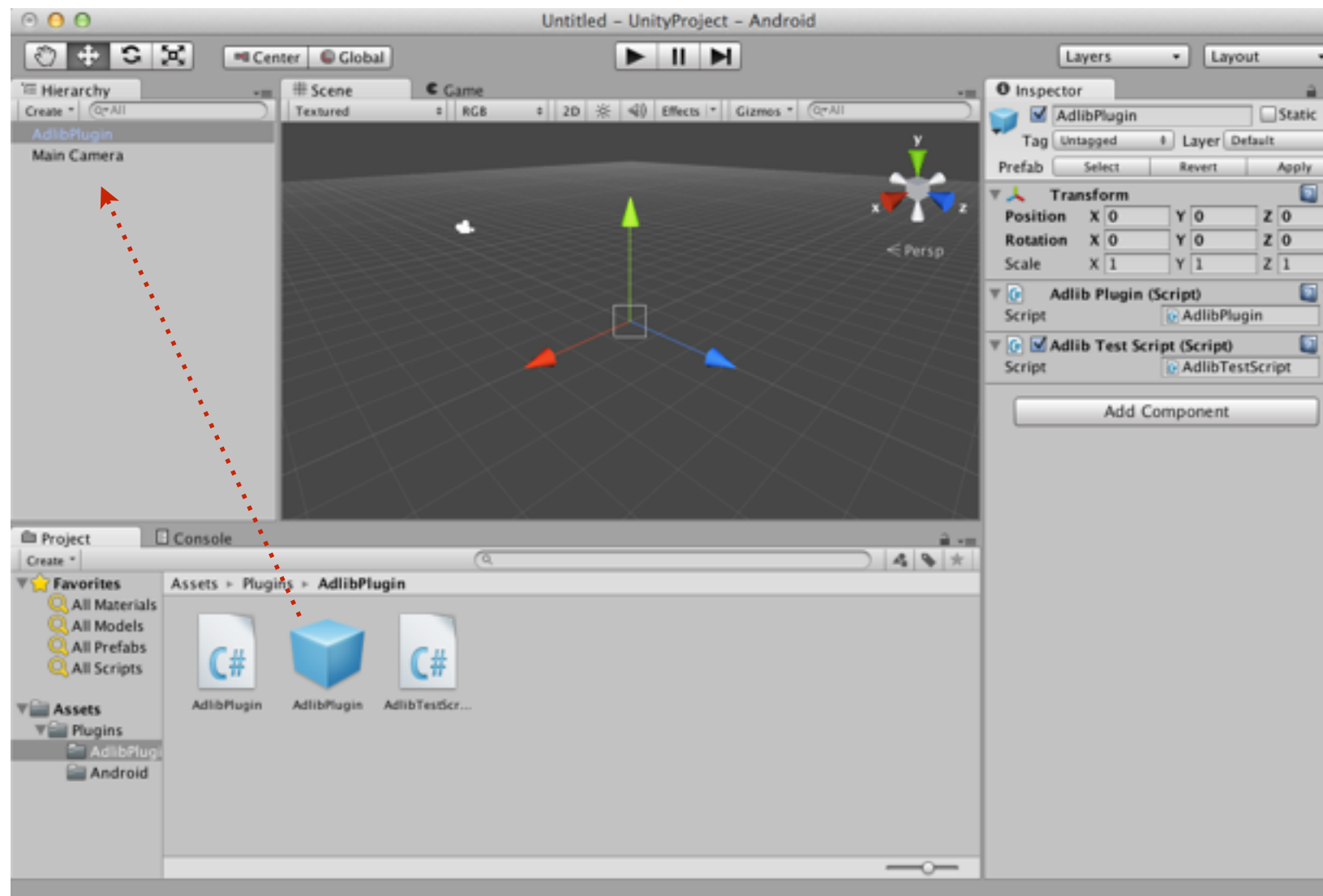
<http://adlibr.com> 의 개발자 지원 메뉴 또는 adlib@mocoplex.com에 남겨주시기 바랍니다.

※중요※

Jar 파일에 리소스를 포함하는 몇몇 플랫폼들은, 해당 리소스를 유니티에서 불러오지 못해 노출이 되지 않을 수 있습니다.

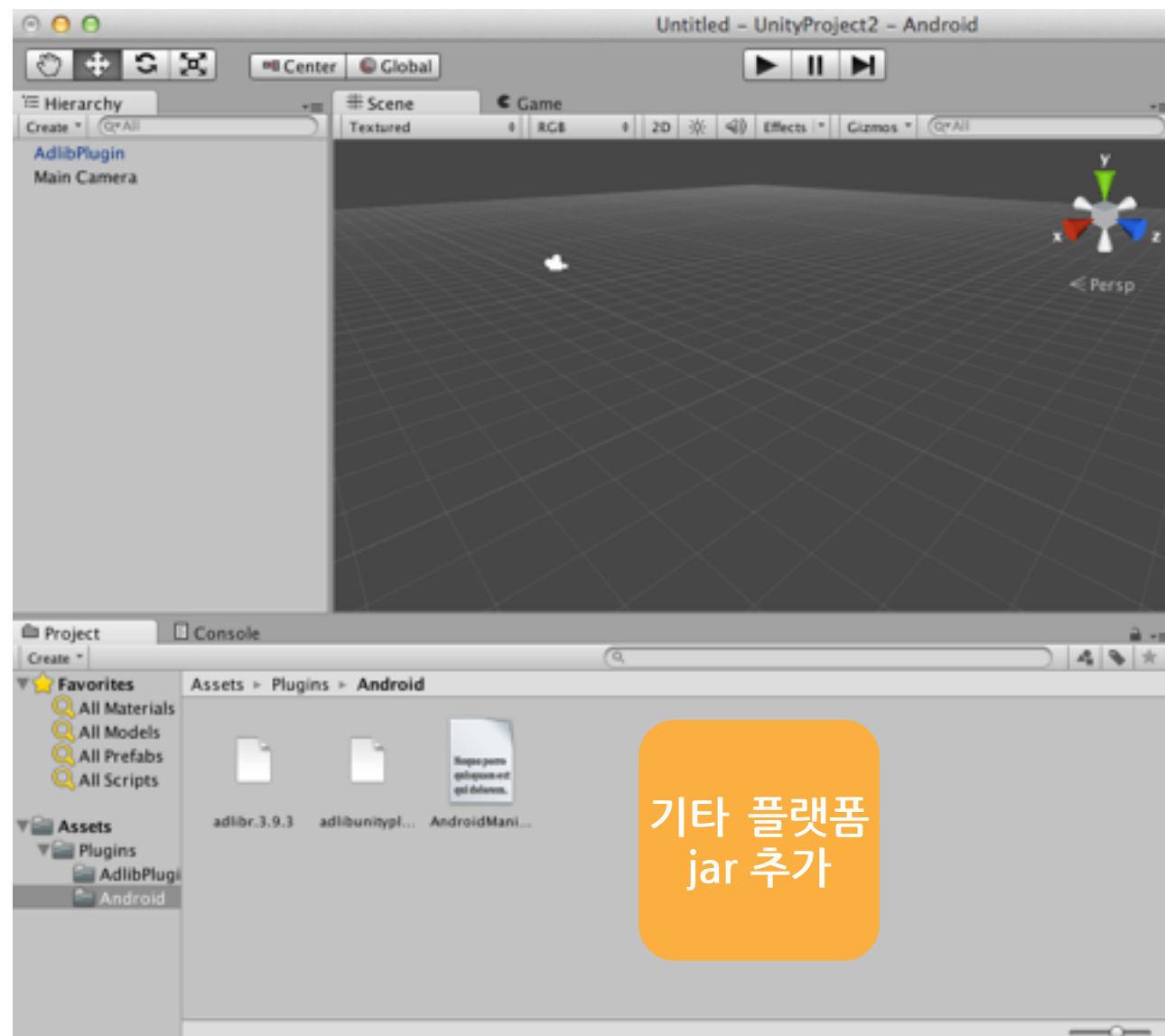
플러그인 import

1. Assets > Import Package > Custom Package 메뉴를 클릭합니다.
2. AdlibUnityPlugin.unitypackage 파일을 선택하고, 모든 항목을 Import 합니다.
3. Assets/Plugins/AdlibPlugin/ 폴더에 있는 AdlibPlugin prefab 을 Unity scene으로 드래그 합니다.



플러그인 적용 #1

1. 안드로이드 빌드 후 Temp/StagingArea 폴더에 있는 AndroidManifest.xml 파일을 복사하여 Assets/Plugins/Android 폴더에 넣습니다.
2. Assets/Plugins/Android/AndroidManifest.xml 에 애드립 및 타 광고 플랫폼들 사용에 필요한 Activity와 퍼미션을 추가합니다.
3. Assets/Plugins/Android 폴더에 애드립 SDK와 기타 광고 플랫폼 SDK들을 넣습니다.



플러그인 적용 #2 - AndroidManifest.xml

```
<!-- 애드립은 Android 2.2(API Level 8) 이상의 환경에서 동작합니다. -->
<uses-sdk android:minSdkVersion="8" />

<!-- 애드립 실행에 필요한 권한 각 플랫폼 별로 요청하는 권한이 모두 다르기 때문에 아래 권한들을 모두 추가하는것을 추천합니다. -->
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<!-- 여기까지 애드립 사용을 위한 필수 권한 -->

<!-- 위치정보를 활용한 보다 최적화된 광고 송출을 위해 필요한 권한입니다. -->
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<!-- 위치정보를 활용한 보다 최적화된 광고 송출을 위해 필요한 권한입니다. -->

<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<!-- 애드립 실행에 필요한 권한 -->

<!-- 애드립 사용을 위해 꼭 추가해주세요. -->
<activity
    android:name="com.mocoplex.adlib.AdlibDialogActivity"
    android:theme="@android:style/Theme.Translucent"
    android:configChanges="orientation|keyboard|keyboardHidden" />

<activity android:name="com.mocoplex.adlib.AdlibWebBrowserActivity"
    android:configChanges="orientation|keyboard|keyboardHidden" />

<activity android:name="com.mocoplex.adlib.AdlibVideoPlayer"
    android:theme="@android:style/Theme.NoTitleBar"
    android:configChanges="orientation|keyboard|keyboardHidden" />
<!-- 애드립 사용을 위해 꼭 추가해주세요. -->

<!-- 광고 클릭을 위해 반드시 아래와 같이 true로 설정되어야 합니다. -->
<meta-data android:name="unityplayer.ForwardNativeEventsToDalvik" android:value="true" />
```

애드립 사용을 위해 위 권한 및 Activity 들이 추가되어야 합니다.

광고 플랫폼별 필요한 권한 및 Activity 들은 각 플랫폼의 SDK 문서를 참조해주세요.

플러그인 적용 #3

애드립 Unity 플러그인에는 애드립 호출을 위한 샘플 스크립트가 포함되어 있습니다.

Assets/Plugins/AdlibPlugin/AdlibTestScript.cs 에 ADLIB_API_KEY 및 각 플랫폼 id 를 입력하시면, 애드립이 동작합니다.

띠배너와 전면배너의 키가 동일할 경우에는 같은 값을 입력해 주시면 되며, 사용하지 않을 플랫폼에 대해서는 삭제하셔도 됩니다.

```
ads.Add(new JsonStringValue("ADLIB","insert_your_ADLIB_API_KEY")); // ADLIB_API_KEY는 반드시 필요
ads.Add(new JsonStringValue("ADAM","insert_your_adam_id")); // adam 띠배너 key 입력
ads.Add(new JsonStringValue("ADAM_INTERSTITIAL","insert_your_adam_interstitial_id")); // adam 전면배너 key 입력
ads.Add(new JsonStringValue("ADMOB","insert_your_admob_id")); // admob key 입력
ads.Add(new JsonStringValue("CAULY","insert_your_cauly_id")); // cauly 띠배너 key 입력
ads.Add(new JsonStringValue("CAULY_INTERSTITIAL","insert_your_cauly_interstitial_id")); // cauly 전면배너 key 입력
ads.Add(new JsonStringValue("ADPOST","insert_your_naverAdPost_id")); // adpost key 입력
ads.Add(new JsonStringValue("INMOBI","insert_your_inmobi_id")); // inmobi 띠배너 key 입력
ads.Add(new JsonStringValue("ADMOB_MEDIATION","insert_your_admob_mediation_id")); // admob 네트워크 key 입력
ads.Add(new JsonStringValue("UPLUS","insert_your_uplus_id")); // uplus 띠배너 key 입력
ads.Add(new JsonStringValue("UPLUS_INTERSTITIAL","insert_your_uplus_interstitial_id")); // uplus 전면배너 key 입력
ads.Add(new JsonStringValue("TAD","insert_your_tad_id")); // tad 띠배너 key 입력
ads.Add(new JsonStringValue("ADHUB","insert_your_adhub_id")); // adhub 띠배너 key 입력
ads.Add(new JsonStringValue("ADHUB_INTERSTITIAL","insert_your_adhub_interstitial_id")); // adhub 전면배너 key 입력
ads.Add(new JsonStringValue("AMAZON","insert_your_amazon_id")); // amazon key 입력
ads.Add(new JsonStringValue("MEZZO","insert_your_mezzoMan_id")); // mezzoMAN key 입력
ads.Add(new JsonStringValue("MILLENNIAL_MEDIA","insert_your_millennialMedia_id")); // millennialMedia key 입력
```

애드립 Unity 플러그인 API

InitializeAdlib(string adsInfo)

애드립 사용에 필요한 플랫폼 별 KEY를 setting하는 메소드로, 최초 한번 호출합니다.

adsInfo 는 JSON 포맷의 string입니다.

adsInfo를 구성하기 위해, 애드립 Unity 플러그인에 기본적으로 JSON for .NET library 가 추가되어 있습니다.

샘플 스크립트대로 형태만 일치 시킨다면 다른 JSON library를 사용하셔도 괜찮습니다.

adsInfo 포맷 예)

```
{“ADLIB” : “애드립API키”, “ADAM” : “아담키”, “ADAM_INTERSTITIAL” : “아담전면키”, “ADMOb” : “애드몹키”, “ADPOST” : “애드포스트키”}
```

메소드 호출 예)

```
JsonObjectCollection ads = new JsonObjectCollection();  
ads.Add(new JsonStringValue("ADLIB", "애드립API키"));  
ads.Add(new JsonStringValue("ADAM", "아담키"));  
ads.Add(new JsonStringValue("ADAM_INTERSTITIAL", "아담전면키"));  
ads.Add(new JsonStringValue("ADMOb", "애드몹키"));  
ads.Add(new JsonStringValue("ADPOST", "애드포스트키"));  
  
AdlibPlugin.InitializeAdlib(ads.ToString());
```

애드립 Unity 플러그인 API

ShowBanner(BannerSize size, bool positionAtTop, int padding)

띠배너를 화면에 노출시킵니다.

| | |
|---------------|---|
| size | 배너 사이즈 설정. BannerSize.Banner (디바이스 width X 50dp 사이즈), BannerSize.Banner_320X50 (320dp X 50dp 사이즈) |
| positionAtTop | true 이면 화면 상단, false이면 화면 하단에 위치. |
| padding | (optional) 상단 또는 하단으로 부터 padding(dp) 만큼 떨어진 곳에 위치. 기본값 0. |

메소드 호출 예)

```
AdlibPlugin.ShowBanner(AdlibPlugin.BannerSize.Banner_320x50, false, 50);
```


애드립 Unity 플러그인 API

HideBanner()

화면에 노출된 띠배너를 숨깁니다.

메소드 호출 예)

```
AdlibPlugin.HideBanner();
```

LoadInterstitialAd()

전면배너를 호출합니다.

메소드 호출 예)

```
AdlibPlugin.LoadInterstitialAd();
```

애드립 Unity 플러그인 API

ShowPopBanner(string frameColor, PopButton btnColor, bool useInAnim, bool useOutAnim, PopAlign btnAlign, int padding)

팝배너를 화면에 노출시킵니다.

| | |
|------------|---|
| frameColor | popBanner의 프레임 컬러 설정. “#RRGGBB” 또는 “#AARRGGBB”. |
| btnColor | popBanner의 Close 버튼 배경색 설정. PopButton.White (흰색 배경의 버튼), PopButton.Black (검정색 배경의 버튼). |
| useInAnim | true 이면 배너가 나타날 때 슬라이드 애니메이션 적용, false이면 애니메이션 없이 배너가 나타남. |
| useOutAnim | true 이면 배너가 사라질 때 슬라이드 애니메이션 적용, false이면 애니메이션 없이 배너가 사라짐. |
| btnAlign | 팝배너 노출 및 애니메이션 적용 기준 position 설정. PopAlign.Left (좌측 기준 정렬), PopAlign.Top (상단 기준 정렬), PopAlign.Right (우측 기준 정렬), PopAlign.Bottom (하단 기준 정렬). |
| padding | 정렬 기준으로 부터 padding(dp) 만큼 떨어진 곳에 위치. |

메소드 호출 예)

```
AdlibPlugin.ShowPopBanner("#ff444444",  
                           AdlibPlugin.PopButton.White,  
                           true,  
                           false,  
                           AdlibPlugin.PopAlign.Bottom,  
                           60);
```

애드립 Unity 플러그인 API

ShowRewardLink(string linkId, RewardLinkAlign linkAlign, int xPos, int yPos)

리워드링크 아이콘을 화면에 노출시킵니다.

여러번 호출하여 여러개의 아이콘을 노출할 수 있습니다.

| | |
|-----------|--|
| linkId | 애드립 홈페이지에서 등록하고 발급받은 리워드링크 ID. |
| linkAlign | 아이콘 노출 기준점 설정. RewardLinkAlign.LeftTop (기준점을 좌측상단으로 설정), RewardLinkAlign.RightTop (기준점을 우측상단으로 설정), RewardLinkAlign.LeftBottom (기준점을 좌측하단으로 설정), RewardLinkAlign.RightBottom (기준점을 우측하단으로 설정). |
| xPos | 기준점으로 부터 x축으로 xPos(dp) 만큼 떨어진 곳에 위치. |
| yPos | 기준점으로 부터 y축으로 yPos(dp) 만큼 떨어진 곳에 위치. |

메소드 호출 예)

```
AdlibPlugin.ShowRewardLink("리워드링크ID",  
                             AdlibPlugin.RewardLinkAlign.RightTop,  
                             50,  
                             60);
```

애드립 Unity 플러그인 API

HidePopBanner()

화면에 노출된 팝배너를 닫습니다.

메소드 호출 예)

```
AdlibPlugin.HidePopBanner();
```

HideRewardLink()

화면에 노출된 모든 리워드링크 아이콘을 숨깁니다.

메소드 호출 예)

```
AdlibPlugin.HideRewardLink();
```

애드립 Unity 플러그인 API

StopAds()

광고클릭, 홈버튼 클릭 등으로 앱이 pause 상태가 되었을 때 애드립 서비스 호출을 중지합니다.

메소드 호출 예)

```
AdlibPlugin.StopAds();
```

RestartAds()

앱이 pause 상태였다가 다시 돌아왔을 때 애드립 서비스를 재개 합니다.

메소드 호출 예)

```
AdlibPlugin.RestartAds();
```

애드립 Unity 플러그인 API

애드립 Unity 플러그인은 다음의 리스너 이벤트를 제공합니다.

```
public static event Action<string> ReceivedInterstitial; // 전면배너 수신 성공 - string은 수신된 플랫폼 이름
public static event Action FailedToReceiveInterstitial; // 전면배너 수신 실패
public static event Action DismissedInterstitial; // 전면배너 닫힘
public static event Action ReceivedPopBanner; // 팝배너 수신 성공
public static event Action FailedToReceivePopBanner; // 팝배너 수신 실패
public static event Action DismissedPopBanner; // 팝배너 닫힘
```

`+=` 연산자를 이용해서 아래와 같이 이벤트를 등록할 수 있습니다.

```
// 리스너 응답을 받을 메소드명을 이벤트에 등록
AdlibPlugin.ReceivedInterstitial += HandleReceivedInterstitial;

// 리스너 응답후 호출부 구현
public void HandleReceivedInterstitial(string platform) {
    print("Received Interstitial Ad : ");
    print(platform);
}
```