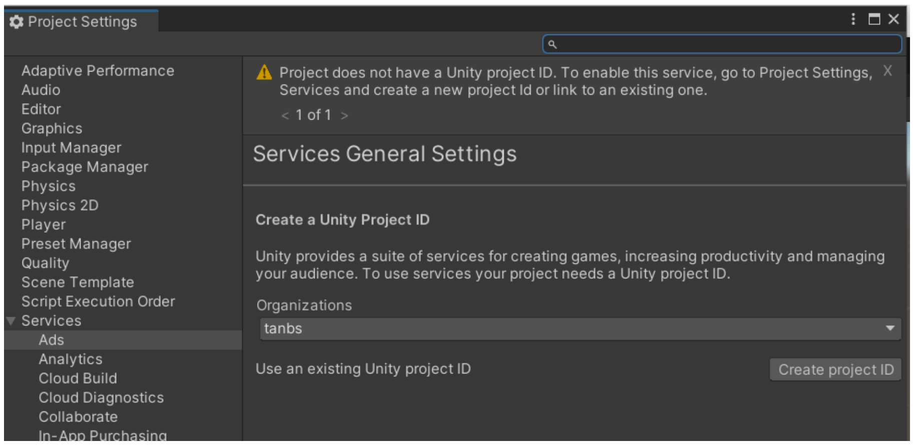
Part 1 – Create Unity Ads

1. Create a new project.
2. In the Unity Editor, select Window > Package Manager to open the Package Manager.
3. Select Ads and create Unity project ID


Project Settings

Adaptive Performance
Audio
Editor
Graphics
Input Manager
Package Manager
Physics
Physics 2D
Player
Preset Manager
Quality
Scene Template
Script Execution Order
Services
Ads
Analytics
Cloud Build
Cloud Diagnostics
Collaborate
In-App Purchasing

Project does not have a Unity project ID. To enable this service, go to Project Settings, Services and create a new project Id or link to an existing one.
< 1 of 1 >

Services General Settings

Create a Unity Project ID

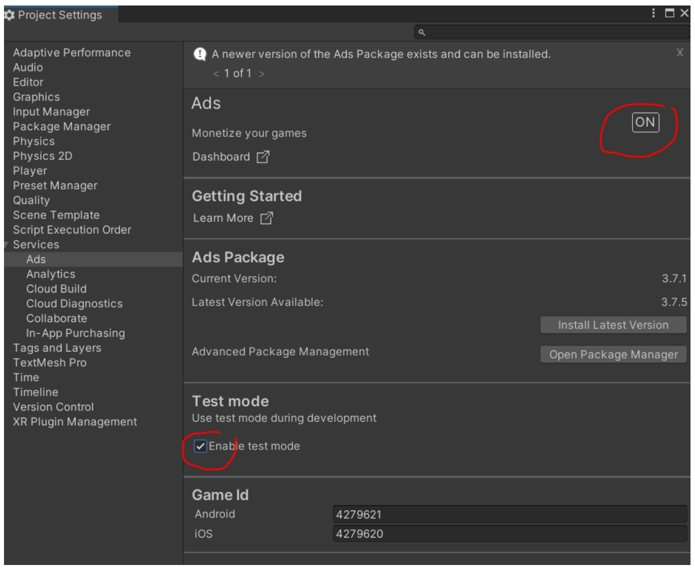
Unity provides a suite of services for creating games, increasing productivity and managing your audience. To use services your project needs a Unity project ID.

Organizations: tanbs

Use an existing Unity project ID

Create project ID

4. Answer "No" for the question "Will this app be primarily targeted to children under age 13"
5. Turn on the Ads and enable Test mode. Lastly Select "Install latest version" to install the sdk for Unity Ads.


Project Settings

Adaptive Performance
Audio
Editor
Graphics
Input Manager
Package Manager
Physics
Physics 2D
Player
Preset Manager
Quality
Scene Template
Script Execution Order
Services
Ads
Analytics
Cloud Build
Cloud Diagnostics
Collaborate
In-App Purchasing
Tags and Layers
TextMesh Pro
Time
Timeline
Version Control
XR Plugin Management

A newer version of the Ads Package exists and can be installed.
< 1 of 1 >

Ads — ON
Monetize your games
Dashboard

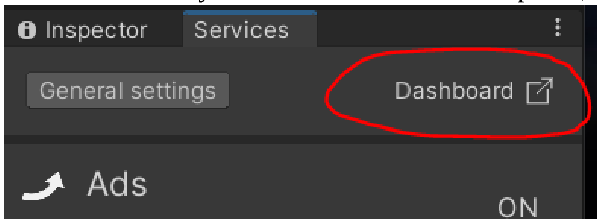
Getting Started
Learn More

Ads Package
Current Version: 3.7.1
Latest Version Available: 3.7.5
Install Latest Version
Advanced Package Management — Open Package Manager

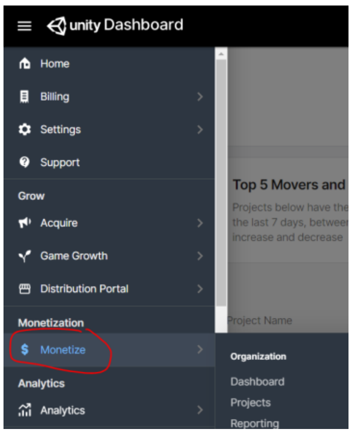
Test mode
Use test mode during development
☑ Enable test mode

Game Id
Android: 4279621
iOS: 4279620

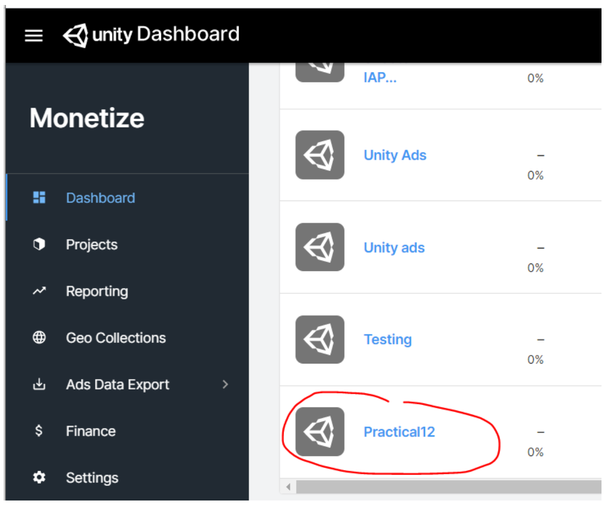
6. From the Unity Build Settings, change the platform to Android.
7. Back to the Unity Editor. From the Service panel, select **Ads> Dashboard**.



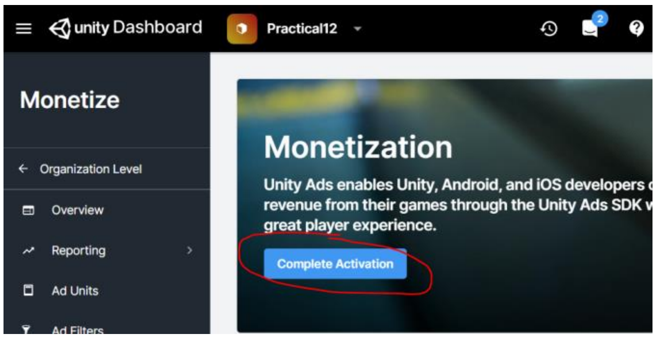
8. From the Dashboard, select Monetize menu.



9. Scroll down from the Monetize dashboard and select the project file

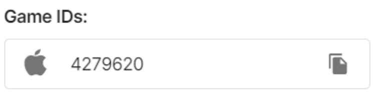


10. Select "Complete Activation"

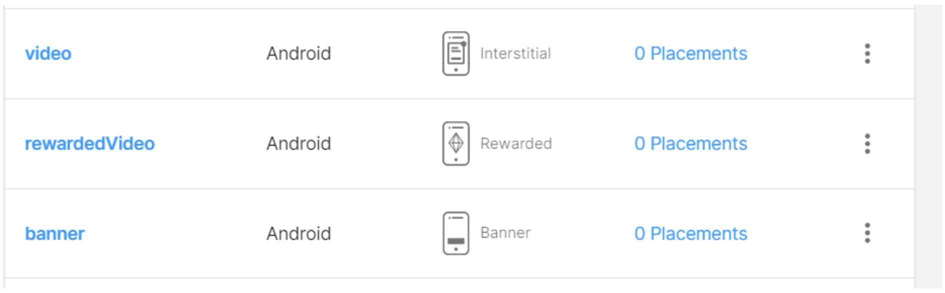


Unity Ads enables Unity, Android, and iOS developers revenue from their games through the Unity Ads SDK great player experience.

11. After the Monetization is activated, Game IDs are created.

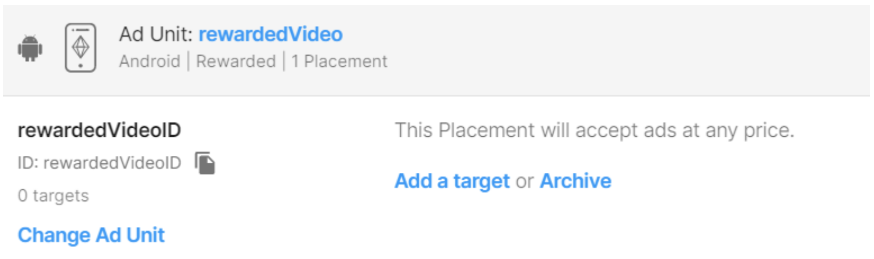


12. From the Ad units page, there are three types of advertisement that can be displayed: Interstitial Ads, Reward Ads and Banner Ads.
13. Create new Ad Unit as following
    a) Ad Unit Name – video, Platform – Android, Ad Format – Interstitial
    b) Ad Unit Name – rewardedVideo, Platform- Android, Ad Format – Rewarded
    c) Ad Unit Name – banner, Platform- Android, Ad Format- Banner



video	Android	Interstitial	0 Placements	⋮
rewardedVideo	Android	Rewarded	0 Placements	⋮
banner	Android	Banner	0 Placements	⋮

14. Create placementID for the three ads by selecting “**Add Placement**”.
    a) video – videoID
    b) rewardedVideo- rewardedVideoID
    c) banner-bannerID

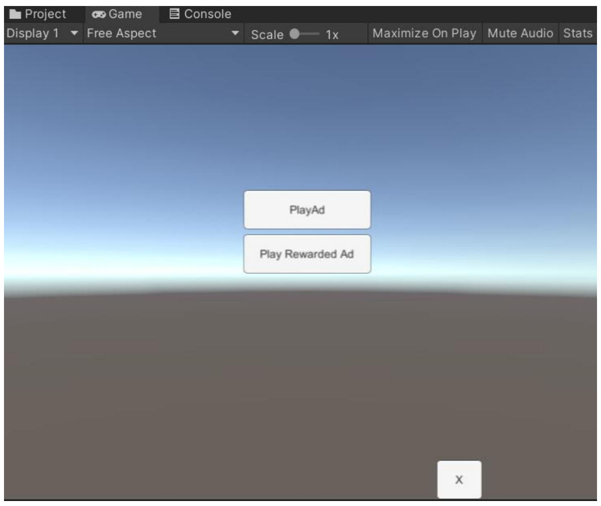


Ad Unit: rewardedVideo
Android | Rewarded | 1 Placement

rewardedVideoID
ID: rewardedVideoID
0 targets
Change Ad Unit

This Placement will accept ads at any price.
Add a target or Archive

15. Back to the Unity Editor. Create UI as following.



16. Select Canvas. Create a new script named “**AdsManager**”. Open the script.
17. Import Advertisement library.

```
using UnityEngine.Advertisements;
```

18. In the Start function, initialize the advertisement by passing the App ID as parameter. Copy App ID from step 11.

```
void Start()
 {
    Advertisement.Initialize("123456");
 }
```

19. Create a new function to play interstitial ads. To do so, create a new function called PlayAd to play the Unity ads named “video” that have created in step 14.

```
public void PlayAd()
    {
        if (Advertisement.IsReady"videoID")) {
            Advertisement.Show("videoID");
        }
    }
```

20. Next, create a new function for Play rewarded video.

```
public void PlayRewardAd()
    {
        if (Advertisement.IsReady("rewardedVideoID"))
        {
            Advertisement.Show("rewardedVideoID");
        }
        else
        {
            Debug.Log("rewarded ad is not ready");
        }
    }
```

21. Next, in order to know whether the reward video has been finish playing, implement a new interface name **IUnityAdsListener**.

```
public class AdsManager: MonoBehaviour, IUnityAdsListener {
……
void Start(){
…
Advertisement.AddListener(this); // add this
}
public void OnUnityAdsDidFinish(string placementId, ShowResult showResult)
    {
        if(placementId== "rewardedVideoID" && showResult == ShowResult.Finished)
        {
            Debug.Log("player should be rewarded");
        }
    }

    public void OnUnityAdsReady(string placementId)
    {
        Debug.Log("ads are ready");
    }

    public void OnUnityAdsDidError(string message)
    {
        Debug.Log("error:" + message);
    }

    public void OnUnityAdsDidStart(string placementId)
    {
        Debug.Log("video started");
    }
```

22. Now its time to create banner ads. From the Hierarchy panel. Create an empty game object named “banner”.
23. From the Inspector panel, create a new script BannerAds. Drag and drop Close button “x” to the “button”.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.Advertisements;
public class BannerAds : MonoBehaviour
{
    public string gameId = "4279621"; //replace with your gameID
    public string placementId = "bannerID"; //replace with your
placement ID
    public bool testMode = true;

    public GameObject button; // to active Close button “x”
    void Start()
    {
        button.SetActive(false);
        // Initialize the SDK if you haven't already done so:
        Advertisement.Initialize(gameId, testMode);
        StartCoroutine(ShowBannerWhenReady());
    }

    IEnumerator ShowBannerWhenReady()
    {
        while (!Advertisement.IsReady(placementId))
        {
            yield return new WaitForSeconds(0.5f);
        }
        Advertisement.Banner.SetPosition(BannerPosition.BOTTOM_CENTER);
        Advertisement.Banner.Show(placementId);
        button.SetActive(true);

    }
}
```

24. Next, add lines below in the Ads manager to deactive Close button “x” after closing the banner. Drag and drop Close button “x” to the “button”.

```
public GameObject button;
 // Start is called before the first frame update
 void Start()
 {
    button.SetActive(true);
    …
 }
```

```
public void HideBanner()
    {
        Advertisement.Banner.Hide();
        button.SetActive(false);
    }
```

25. Further reading
    https://docs.unity3d.com/Packages/com.unity.ads@3.2/manual/MonetizationBannerAdsUnity.html

Testing

Prior to publishing your game, enable test mode by following these steps:

1. From the Operate tab of the Developer Dashboard, select your Project.
2. Select **Settings** > **Project Settings** from the left navigation bar.
3. Scroll down to the **Test mode** section and click the edit button on a platform (Apple App Store or Google Play Store), check **Override client test mode** box, then select the Force test mode **ON** radio button.

In the Unity Editor, click the play button to run your Project and test your ads implementation.

Note: You must enable test mode before testing ads integration, to avoid getting flagged for fraud.

Update Score after watch reward video

1. From the Hierarchy panel, create a Text.
2. Create a new script named “rewardScore”.
3. Declare Text and score variable.
4. Create a function OnRewardAdSuccess() to return the score.
5. Open AdsManager. Modify this method to update the score.

```
  public void OnUnityAdsDidFinish(string placementId, ShowResult
showResult)
   {
      if (placementId == "rewardedVideoID" && showResult ==
ShowResult.Finished)
      {
         //Debug.Log("player should be rewarded");
         FindObjectOfType<rewardScore>().OnRewardAdSuccess(); //add this

      }
```

Further reading

https://blog.unity.com/games/7-steps-to-grow-your-game-with-paid-user-acquisition

https://learn.unity.com/course/growing-your-mobile-game?_gl=1*3q74t6*_ga*MTI1MTE5MDIwMy4xNjE3MTgwODA1*_ga_1S78EFL1W5*MTYyOTg4MjA0OC43OC4xLjE2Mjk4ODIyNzUuNTI.&_ga=2.118154957.550950133.1629882049-1251190203.1617180805

Terminology of monetization - https://unityads.unity3d.com/help/resources/statistics#terminology

Revenue and payment

https://unityads.unity3d.com/help/resources/revenue-and-payment

https://forum.unity.com/threads/average-income-per-watched-video-unity-ads.323588/

Unity Auction

https://unityads.unity3d.com/help/programmatic/exchange-partners

Unity Campaign

https://unityads.unity3d.com/help/advertising/configuring-campaigns