

# Supporting API level 26 for Unity

## Version 3

Last year we [announced](#) that starting from August 2018 **all new** apps and games submitted to the Google Play Store will require a target API level of 26 or above (Android 8.0 Oreo), including games based on Unity. From November 2018 this will also apply to updates to existing apps. More information available [here](#).

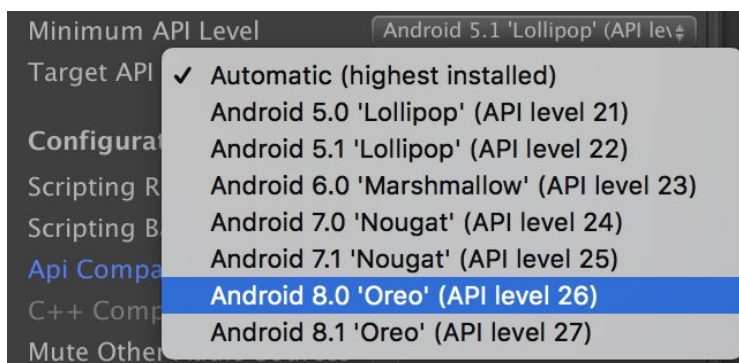
These approaches can help you to set the targetSdkVersion to 26 in your Unity IDE, however note that actual support for API 26 behavior changes depends on the version of the Unity engine you are using, as well as any plugins or SDKs. Test your game on API 26 before publishing especially if you have altered the Android manifest manually. In general, Unity's recommendation is to migrate to supported versions like Unity 2017.4 (Long Term Support release) as soon as possible. Please note that Unity 5.6 and earlier versions are no longer supported. More info here: <https://unity3d.com/unity/qa>

## Unity 5.6.6 and Beyond

Unity officially supports API level 26 on 5.6.6 and beyond.

To set the Target API level in Unity:

1. Open *Build Settings / Android / Player Settings*.



2. Set Target API Level to 26.
3. Build APK

# Unity 4.3.x to 5.6.5

Prior Unity versions use a command that is deprecated in the latest Android SDK tools, which breaks the build process. The workaround is to replace the tools directory with the one from the previous version of the Android SDK. There are two known ways to build a Unity project:

## Using Unity build menu

1. Install Android SDK tools version 25.1.7 ([Windows](#), [Mac](#), [Linux](#))
  - a. Unpack in: `~/$user/sdk_unity/`
2. Install dependencies
  - a. `cd ~/$user/sdk_unity/tools`
  - b. `./android`
  - c. In the UI select:
    - i. Build-tools 26
    - ii. Platforms 26
  - d. In Unity → Preference → External Tools, set the Android SDK path to:  
`~/$user/sdk_unity/tools`
3. Build the APK
4. Verify that `android:targetSdkVersion="26"` in the APK's AndroidManifest.xml
  - a. Open Android Studio 3+
  - b. Profile or debug APK
  - c. Load APK from Unity & verify `android:targetSdkVersion="26"`

## Exporting Android project

When the Internal Build workaround is not applicable, exporting the project, modifying the manifest, importing the project in Android Studio and building is a viable alternative.

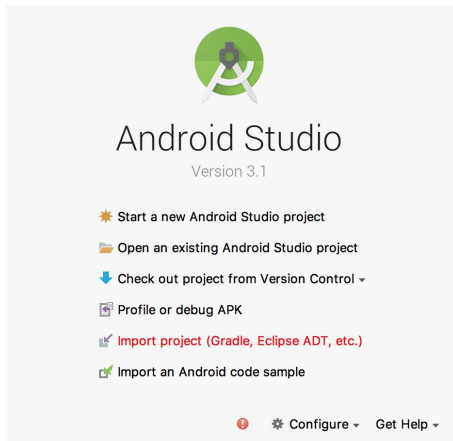
### Export a Unity 4.x/5.x project to ant

1. Install Android SDK tools version 25.1.7 ([Windows](#), [Mac](#), [Linux](#))
  - a. Unpack in: `~/$user/sdk_unity/`
2. Install dependencies:
  - a. `cd ~/$user/sdk_unity/tools`
  - b. `./android`
  - c. In the UI select:

- i. Build-tools 23.0.3
  - ii. Platforms 22
3. In Unity → Preference → External Tools, set the Android SDK path to:  
`~/$user/sdk_unity/tools`
4. Export the Android project
  - a. If you get an error looking like: *Failed to Copy File / Directory from '/Applications/Unity/PlaybackEngines/AndroidPlayer/Data/unity default resources' to 'Temp/StagingArea/Data/unity default resources'.*  
**Fix:** Open your assets folder and delete the “Temp” folder

### Modify the exported project to target API level 26

1. Open AndroidManifest.xml in the exported project folder
2. Modify the line:  
`<uses-sdk android:minSdkVersion="9" android:targetSdkVersion="22" />`  
To:  
`<uses-sdk android:minSdkVersion="9" android:targetSdkVersion="26" />`
3. Open Android Studio, select Import project:



4. Build APK

## Unity 4.2.x and prior

At the moment there is no reliable workaround that we can recommend for Unity versions 4.2 and prior. Some developers have had success manually adjusting the `targetSdkVersion` of their compiled APK, ensuring that any API changes are handled via reflection or JNI.