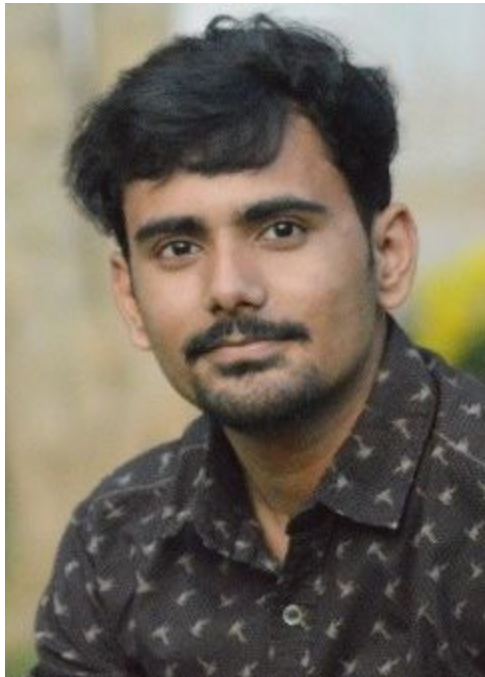


# Integrating a Unity to a Native Android App



Angelhia De Fiesta

EDITED BY VIVEK KUMAR YADAV on 19/4/2020



<https://www.linkedin.com/in/vivek-kumar-yadav-02848a145/>

## **The Goal:**

- Create a native Android app that can launch and start a Unity Module / Project
- Pass some data from native Android app to unity app

## **Requirements:**

- Unity 3D (tried on 2018.4.17f)
- Android Studio (tried on 3.6.3)
- Knowledge in Unity development
- Knowledge in Android development

## Steps:

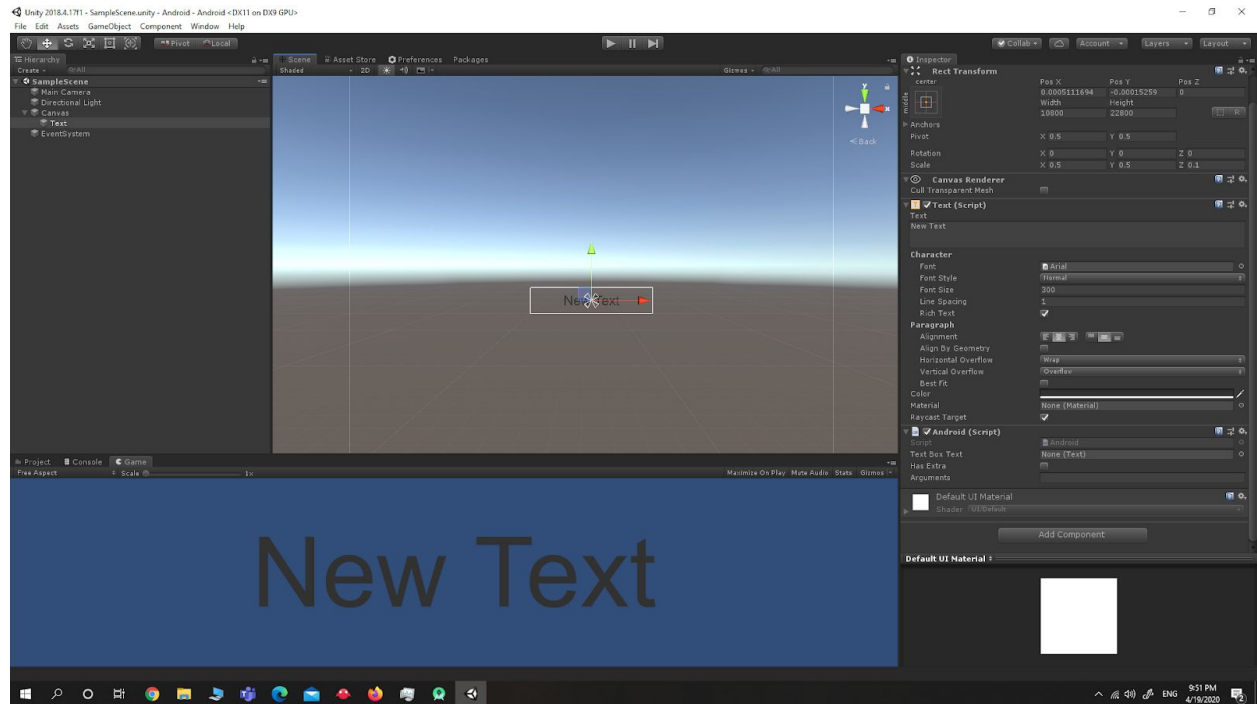
### 1. Create your Unity Project

Open Unity and do your stuff there. Since our Unity app is receiving some data from our Android app, add a Text on your scene which will display the received data. Below code handles the logic of receiving and displaying the data:

```
// Use this for initialization
void Start () {
    TextBoxText = gameObject.GetComponent<Text>();
    AndroidJavaClass UnityPlayer = new
    AndroidJavaClass("com.unity3d.player.UnityPlayer");
    AndroidJavaObject currentActivity =
    UnityPlayer.GetStatic<AndroidJavaObject>("currentActivity");
    intent = currentActivity.Call<AndroidJavaObject>("getIntent");
    hasExtra = intent.Call<bool> ("hasExtra", "arguments");
    Debug.Log("start");
}

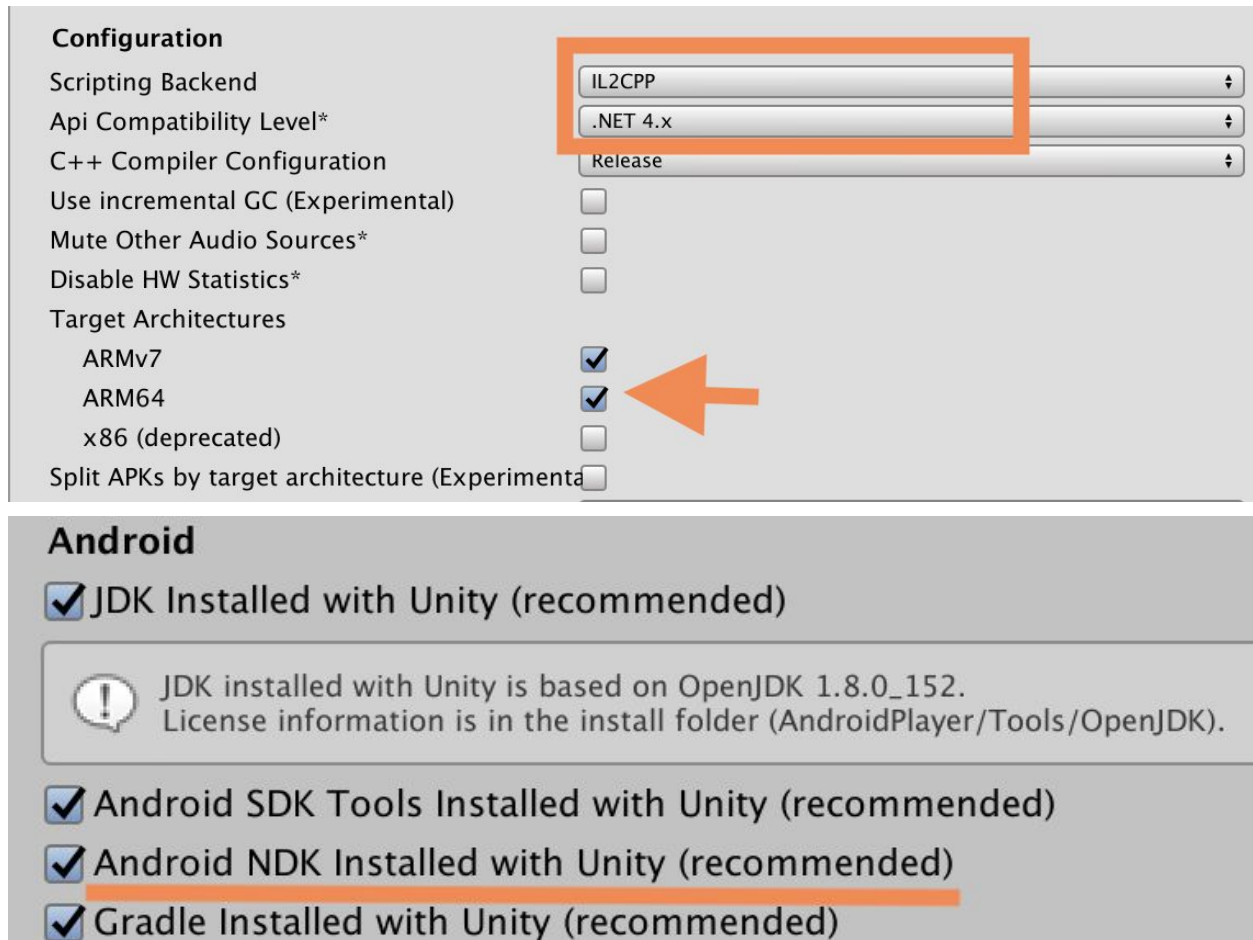
Update is called once per frame
void Update () {
    if (hasExtra) {
        Debug.Log("has extra");
        extras = intent.Call<AndroidJavaObject> ("getExtras");
        arguments = extras.Call<string> ("getString", "arguments");
        TextBoxText.text =arguments;
        Debug.Log(arguments);
    } else {
        TextBoxText.text = "No Extra from Android";
        Debug.Log("no extra");
    }
}
```

}  
}

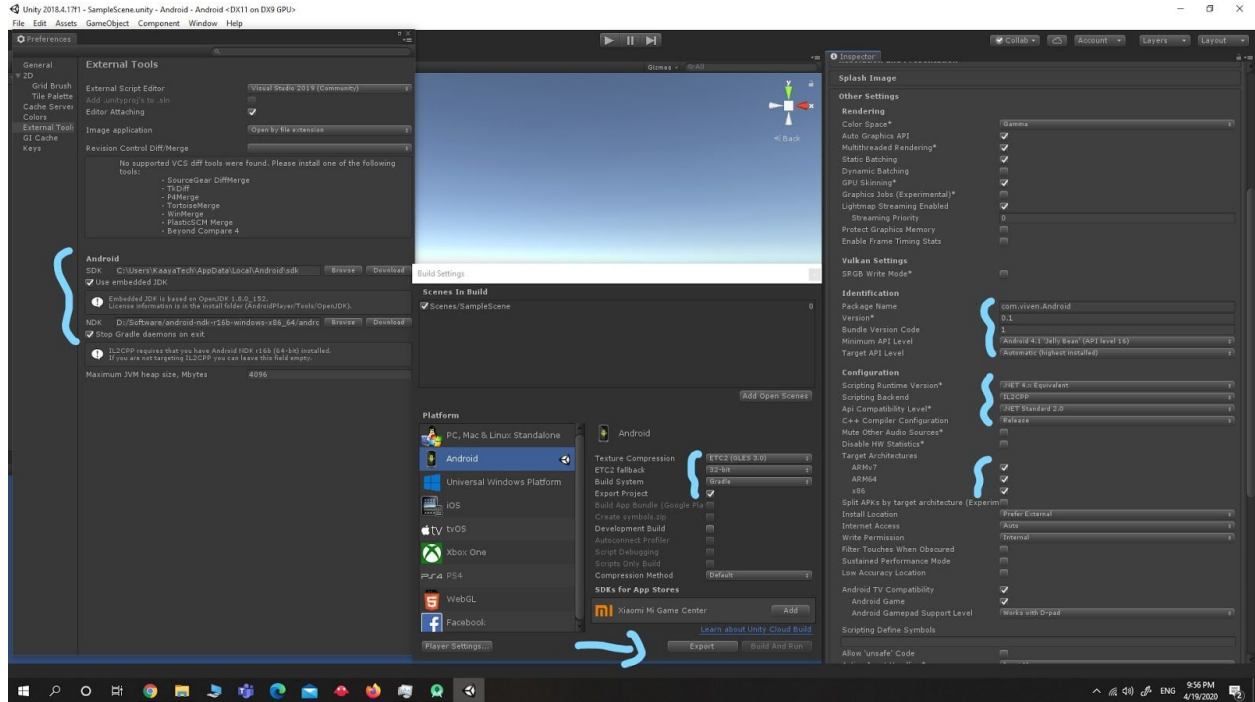


## 2. Export Project

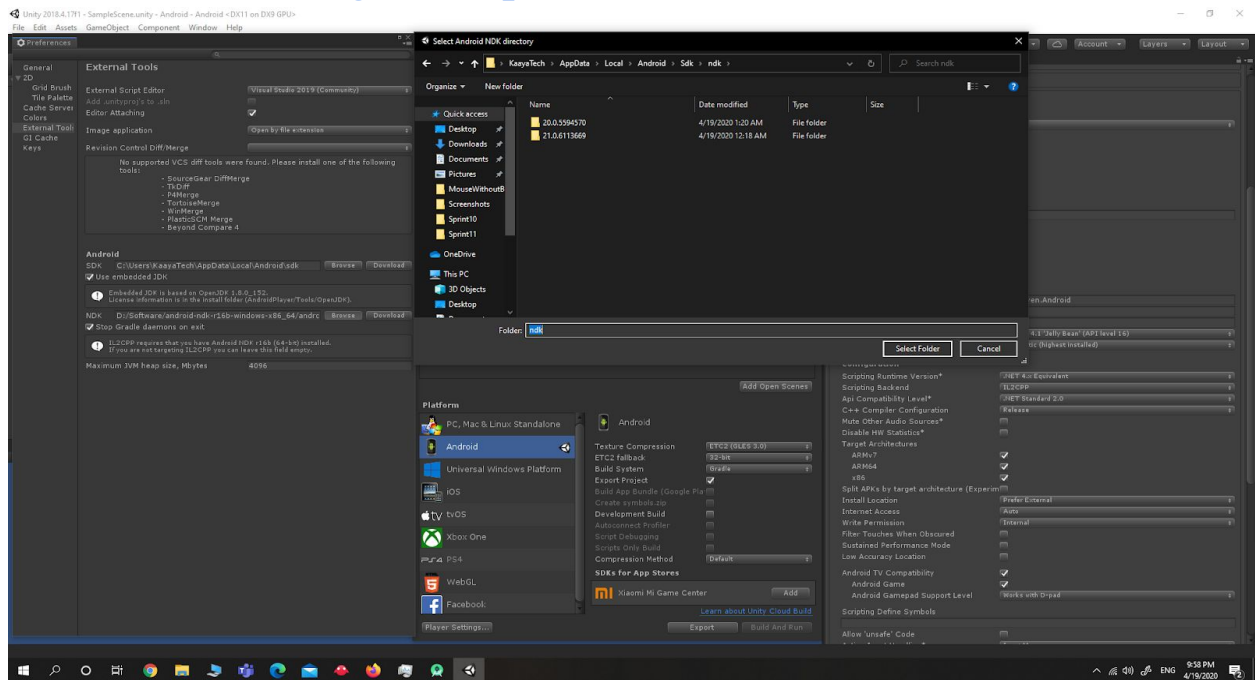
- Click File > Build Settings
- Platform Section > Android
- Change Texture to ETC2
- Change Build System to Gradle
- Click on Player Setting button > Inspector > Other Setting > (Add your package name)

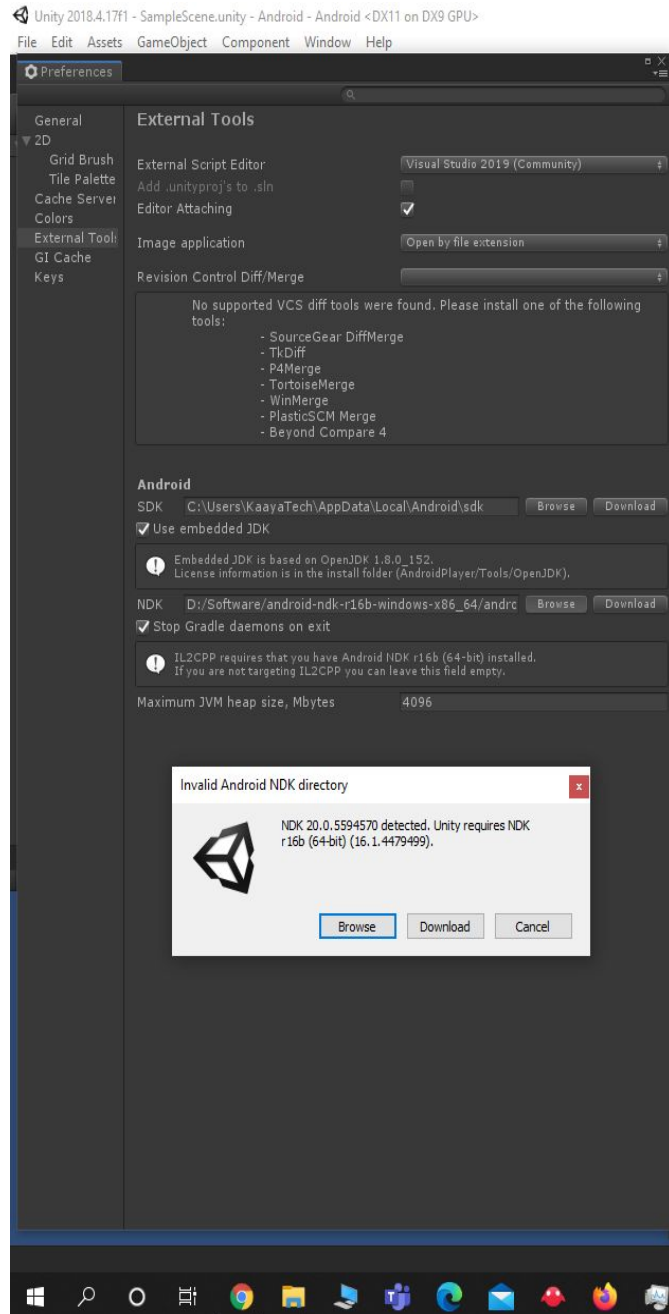


- Mark “Export Project”
  - Now locate where you exported your project
- This is how it should look at the end.*



You will be needing an ndk do rem to download and install one(latest one)(go to preference → external tool and press the download option) but there is a diff b/w the android studio ndk and unity ndk you can see below if u will try to use the same ndk then this error may turn up.





### 3. Convert Unity project to an android library project

— Import the exported Unity project (from step 2) to Android Studio

*Open Android studio and go to file open → and locate the location of the exported folder from the window and open it. You*

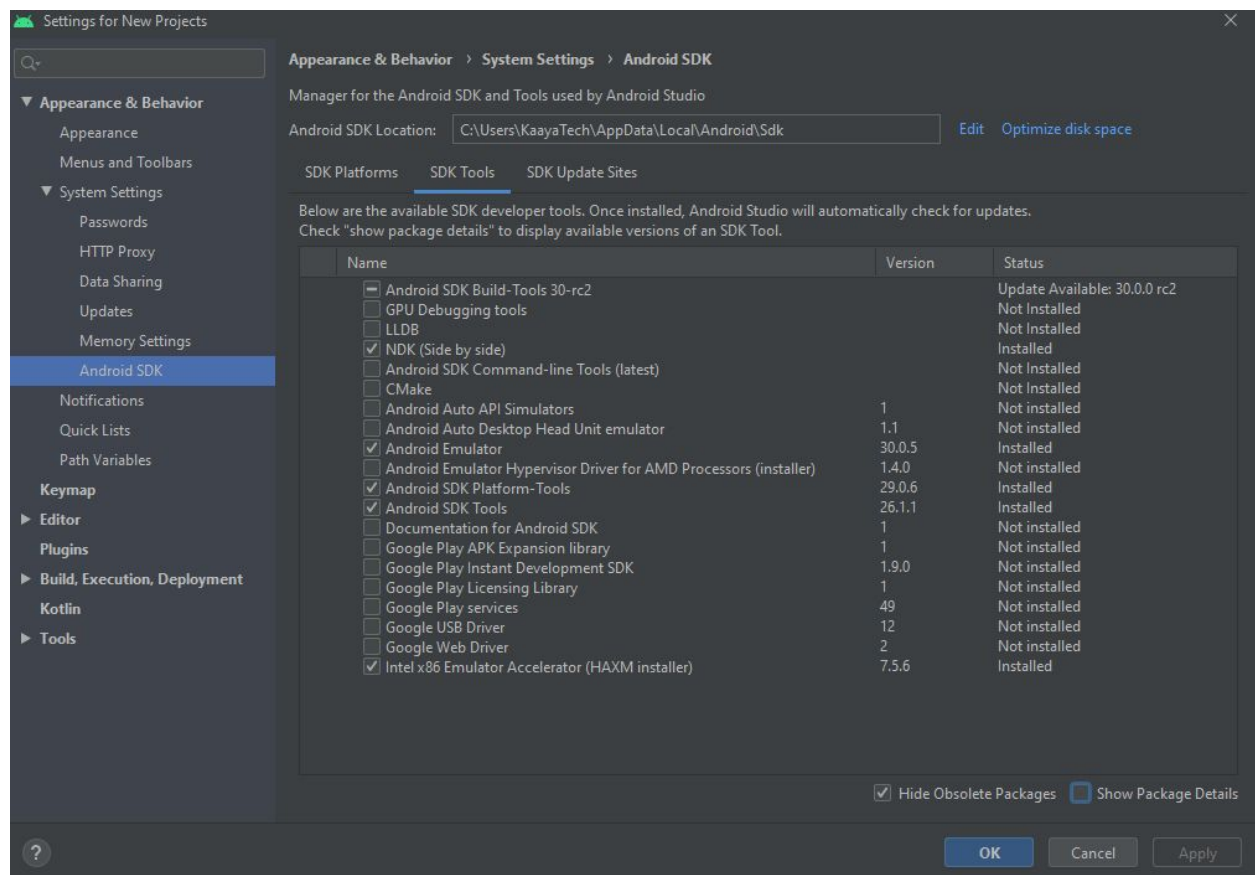
will automatically see an android logo to confirm it. There might be a gradle build error just click resolve and the issue will go away.

→ go to help → update and update your android studio.

→ now install NDK (side by side)

→ go to tools SDK manager click on sdk tool there u will see the NDK side by side just download it.

→ it might so happen that during the build time u may see an error like NDK not found don't worry just click on it it will automatically download the required NDK again.



— Open manifest and remove the intent code:

```
<!--<intent-filter>-->
```

```
<!--<action android:name="android.intent.action.MAIN" />-->
```

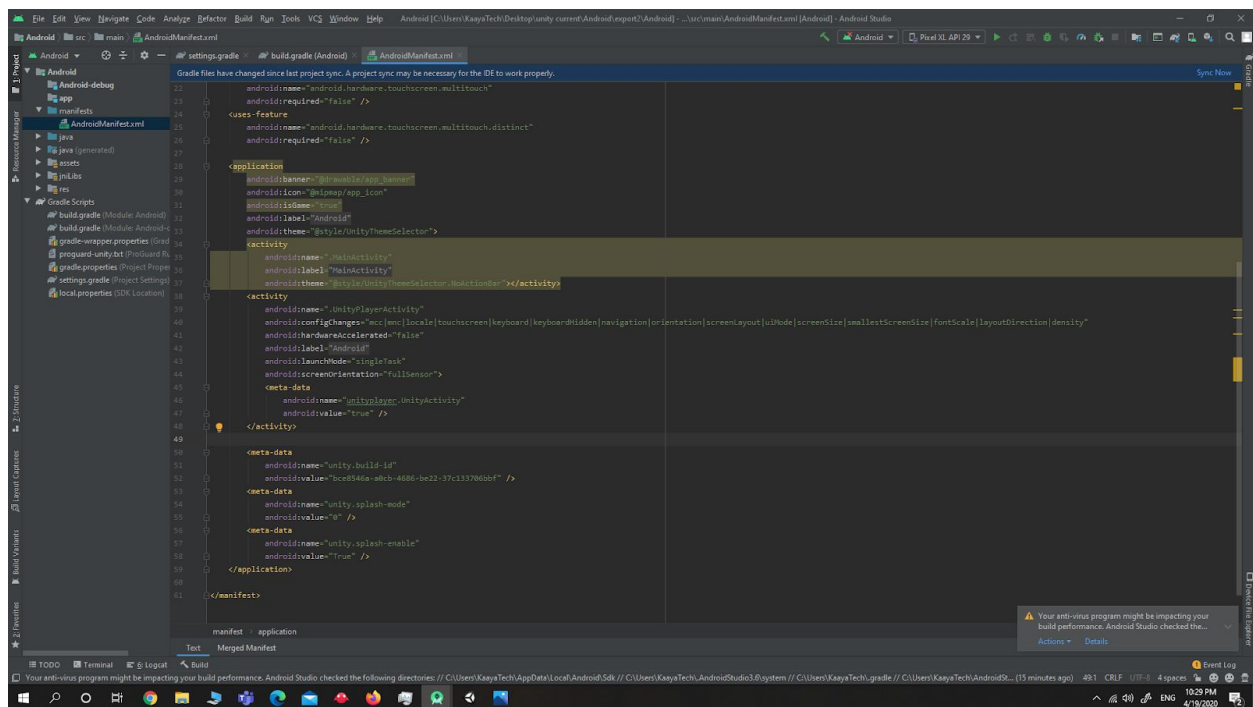


```
<!--<category
android:name="android.intent.category.LAUNCHER"/>-->

<!--<category
android:name="android.intent.category.LEANBACK_LAUNCHER" />-->

<!--</intent-filter>-->
```

Delete from here and press sync now as seen in the top after making any change u have to sync it. See picture for reference



- Open app level build.gradle, change  
apply plugin: 'com.android.application'  
to

```
apply plugin: 'com.android.library'
```

- Delete applicationId

```
defaultConfig {
    applicationId "avdf.com.androidunity_sending" <-- delete this
                                                    line
}
```

```
minSdkVersion 15
targetSdkVersion 25
```



```
versionCode 1
versionName "1.0"
}
```

*And comment out*

```
/* bundle {
    language {
        enableSplit = false
    }
    density {
        enableSplit = false
    }
    abi {
        enableSplit = true
    }
}*/
```

*Do a sync*

*Then press make project or build(apk) you will get a aar file in the said location*

- Build the project, and take the AAR file aside.

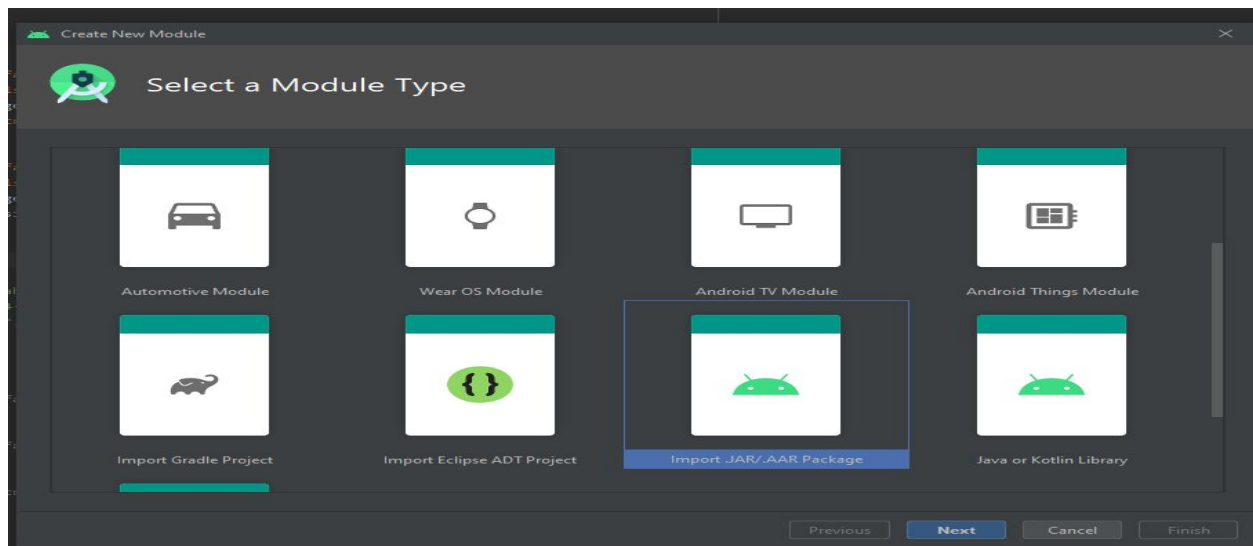
— AAR file should be located in:

PROJECT\_FOLDER\build\outputs\aar

#### **4. Integrate this aar library to your Android App**

— Create a new *android project or open the one in which u want to integrate one* native Android app: Add New Module: File → New Module → Import JAR/AAR Package, and select your aar

file.



— Add a dependency to your app/build.gradle:

```
dependencies {
```

```
    implementation project(":your_aar_file_name") by default its android-debug do rem to put this before it ":"
```

```
}
```

## 5. Edit your Main Activity(go to youtube and watch a video how to create a button in android studio)

```
Intent intent = new Intent(this, UnityPlayerActivity.class);
intent.putExtra("arguments", "data from android");
startActivity(intent);
```

*This is how your MainActivity.java may look*

```
package com.viven.unity_android;
```

```
import androidx.appcompat.app.AppCompatActivity;
```

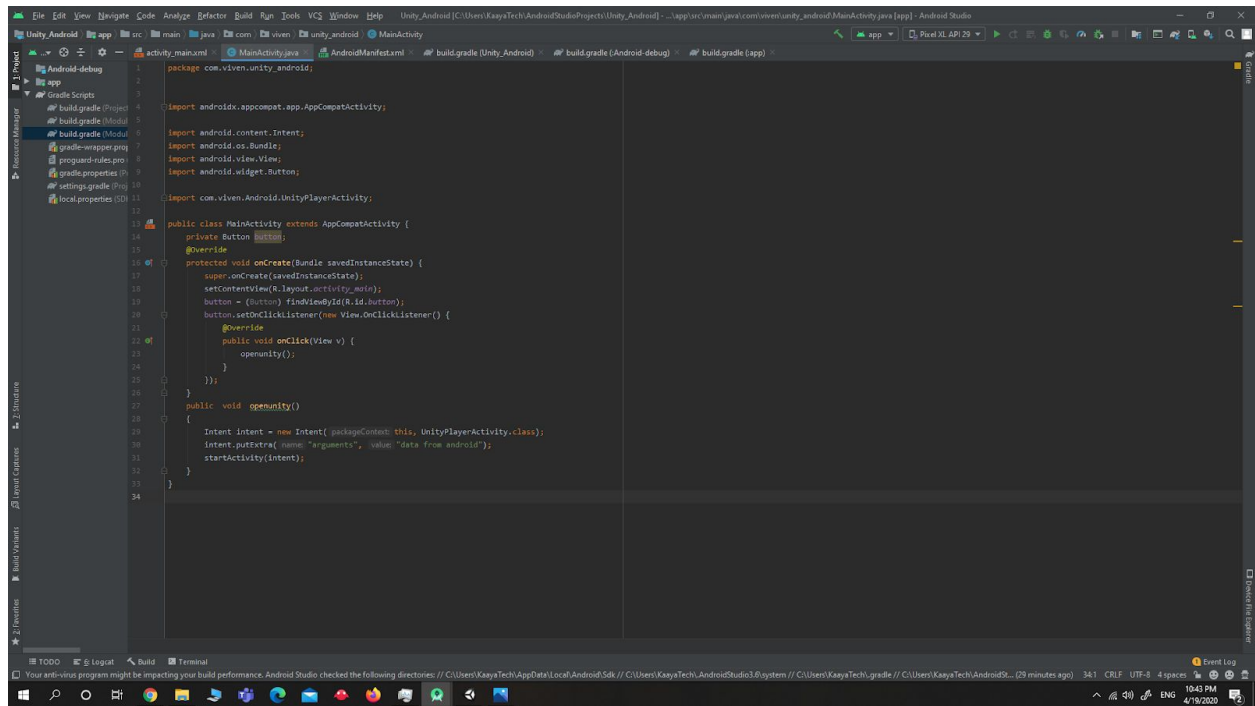
```
import android.content.Intent;
```

```
import android.os.Bundle;
```

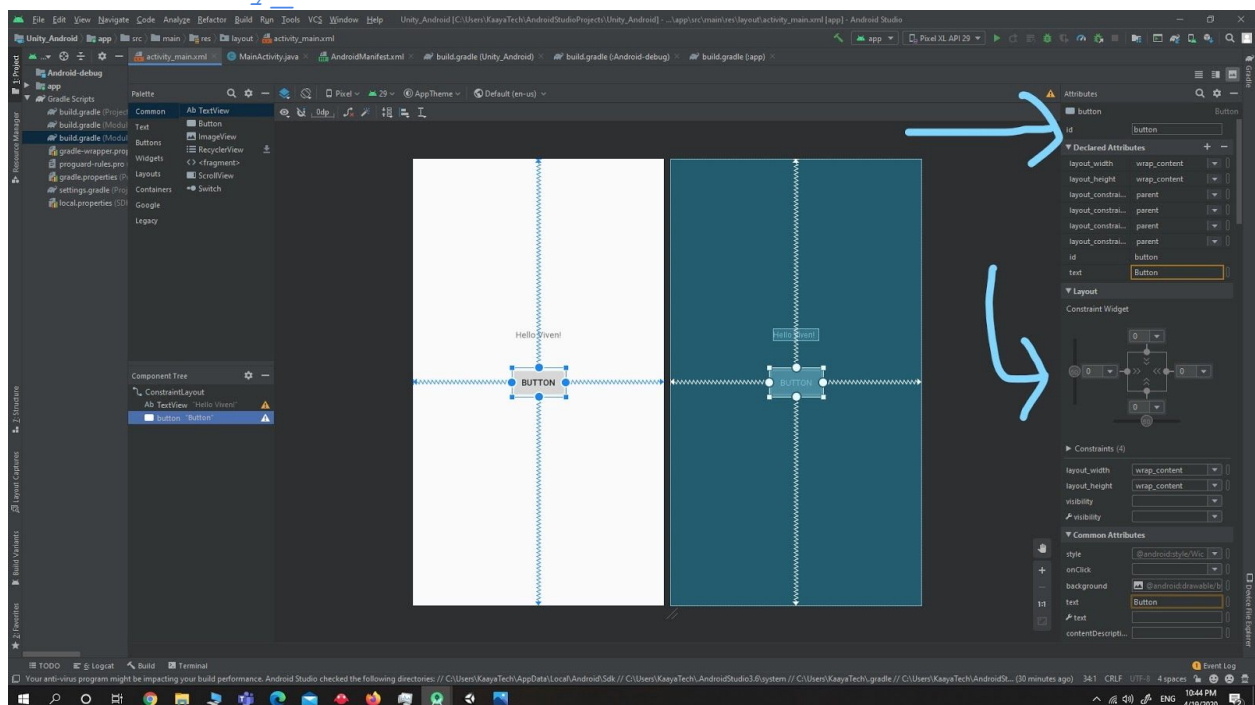
```
import android.view.View;
import android.widget.Button;

import com.viven.Android.UnityPlayerActivity;

public class MainActivity extends AppCompatActivity {
    private Button button;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        button = (Button) findViewById(R.id.button);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                openunity();
            }
        });
    }
    public void openunity()
    {
        Intent intent = new Intent(this, UnityPlayerActivity.class);
        intent.putExtra("arguments", "data from android");
        startActivity(intent);
    }
}
```

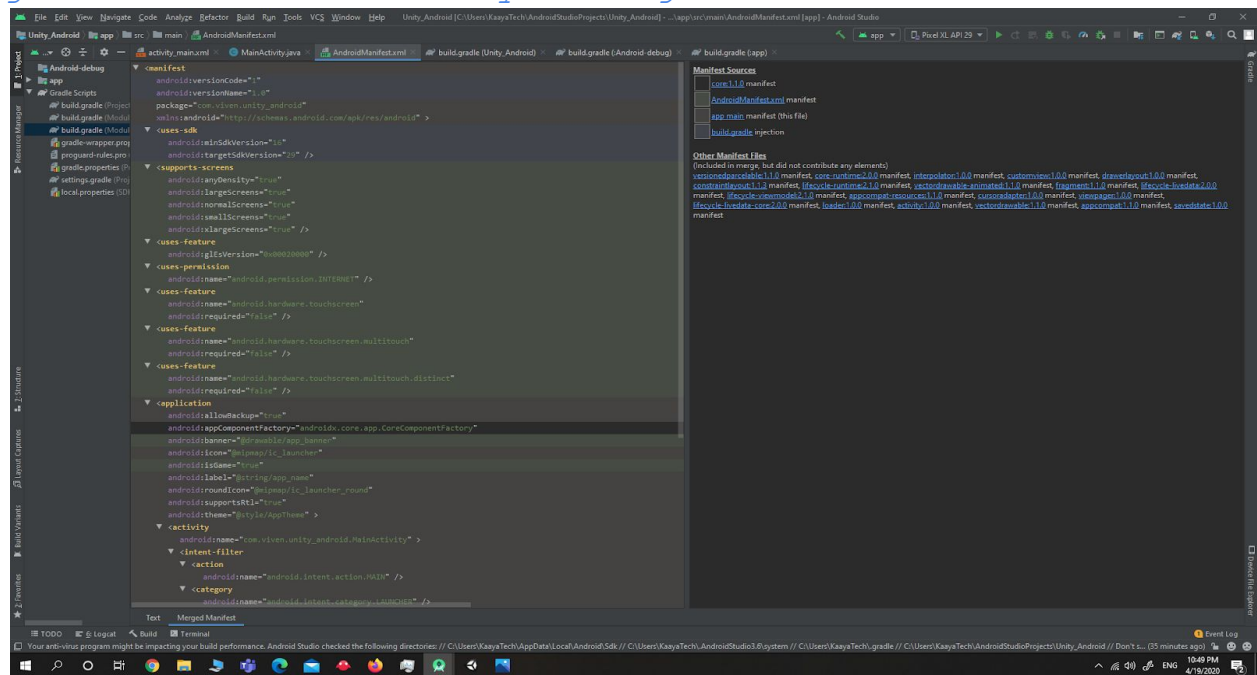


And activity\_main.xml look like this.



If you find any issue with manifest or merger then open AndroidManifest.xml and down belo click on Merged

Manifest and there u will see some issues written over just click on them they will get resolved.



## 6. Build and install your apk.

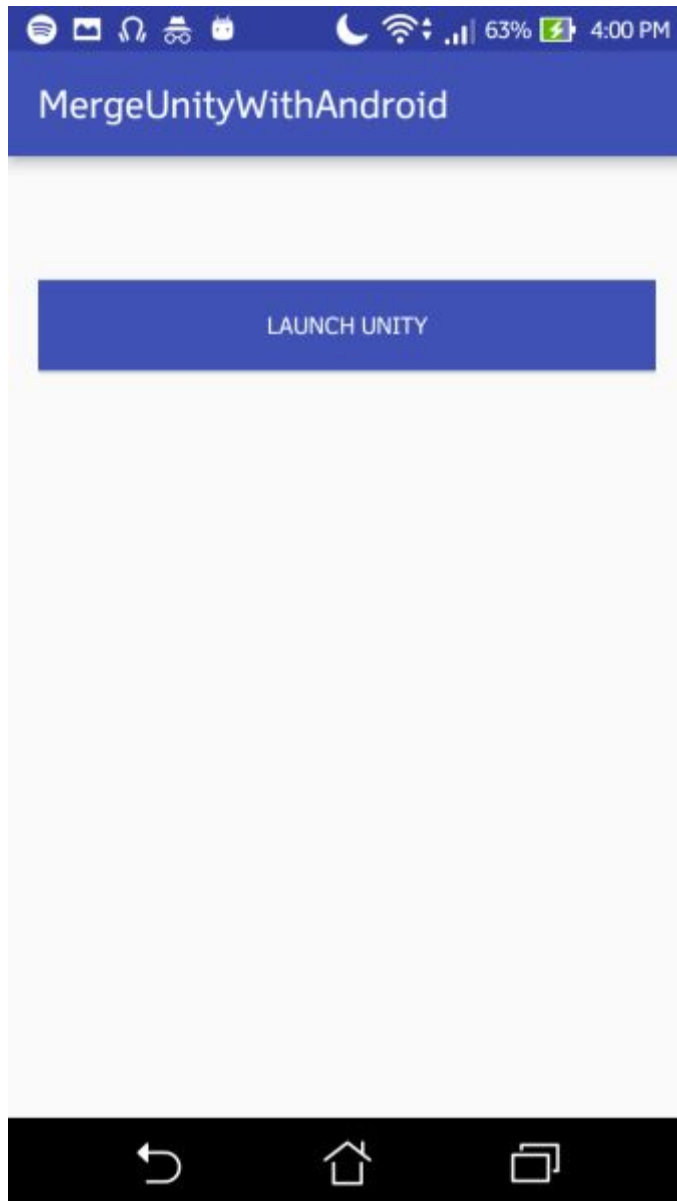
Screenshots:

Clicking “Launch Unity” button will launch your Unity App.

final\_unity

BUTTON

Hello World!




Received data from Android native app should be displayed on your Unity App:





data from android



data from android

Source code:

[Android native](#)

[Unity](#)

Demo:

[apk](#) (a bit customized as EditText is added to the demo apk)

THANK YOU

!!!!!!!!!!!!!!!!!!!!!!  
.....