How to use Jar package in Unity project on Pico device

This document mainly introduces three approaches that how to write and call methods in a Jar package, which can be used in Unity project on Pico device.

Method 1: Inheriting from Class MainActivity

1. Writing Jar Package

1. Import vractivity.jar in SDK assets/plugin/android path in app->libs, and right click vractivity.jar to select add as library

2) Inherit from UnityPlayerNativeActivityPico class, add public static Activity unityActivity = null; in method onCreate(), add unityActivity = this;

3. Remove the line setContentView();



1. Add the following code on the app-> build.gradle root node,

task createJarDebug(type: Jar,dependsOn: ['build']) {  
 "test"  
 from 'build/intermediates/javac/debug/compileDebugJavaWithJavac/classes'  
 include('\*\*')  
 exclude('\*\*/R.class')  
 exclude('\*\*/R$\*.class')

exclude('\*\*/BuildConfig.class')  
}

//需要混淆时添加以下代码

task proguard(type: proguard.gradle.ProGuardTask, dependsOn: buildJar) {  
// 输入路径  
 injars "build/libs/test.jar"  
// 输出路径  
 outjars "lib/testProguard.jar"  
// 添加配置信息  
 configuration 'proguard-rules.pro'  
}

5. Enter gradlew createJarDebug in Terminal

You need to generate the conflation package by typing gradlew createJarDebug proguard

If prompted build success, the package succeeds

Jar package path: app/build/libs/test.jar

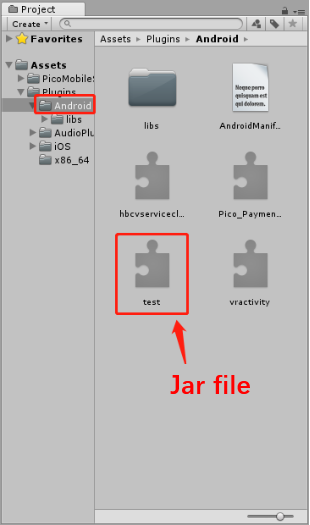
Copy to Unity engineer and tell them package name. Class name, interface name

For example, in step 3 com. Paix. Jartest. MainActivity void startJar ()

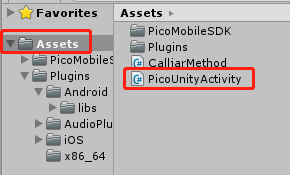
2. Call the Jar package

1. Export the Jar package and remember the package name of the Jar package. Class name, method name

2. Copy the Jar package to the Assets/Plugins/Android/ directory



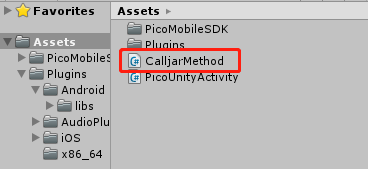
3. Copy the Jar package to the Assets/Plugins/Android/ directory

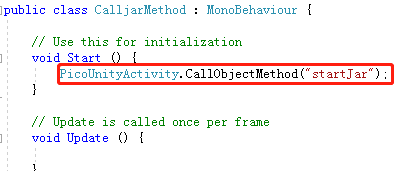


1. Modify the packageName property of the CurrentActivity() method in picounityactive. cs to the packageName of the Jar



1. In the script file, call the Jar package method by passing in the method name through the static method of PicoUnityActivity (see picounityactivity.cs for some overloaded methods).

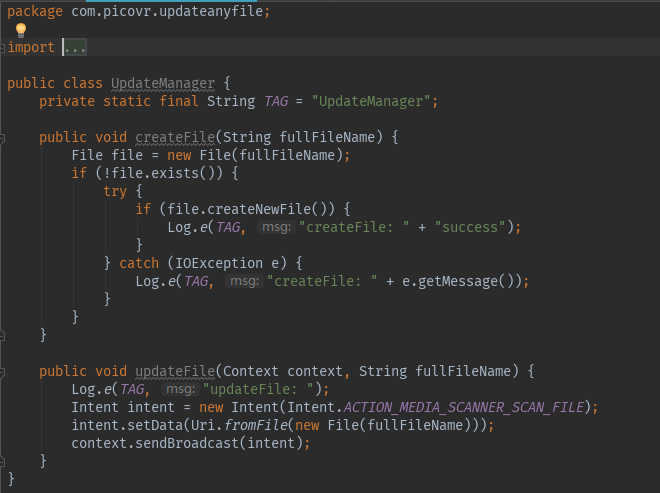




## 2. Non-inheriting the main Activity method

## Export jar file

1. Create a new class and write your own method (here's how to create the file and update the media library



2. Add the following code on the app-> build.gradle root node, which is the same level as android and dependencies

task createJarDebug(type: Jar,dependsOn: ['build']) {  
 baseName "test"  
 from 'build/intermediates/javac/debug/compileDebugJavaWithJavac/classes'   
 include('\*\*')  
 exclude('\*\*/R.class')  
 exclude('\*\*/R$\*.class')  
 exclude('\*\*/BuildConfig.class')  
 exclude('\*\*/MainActivity.class')  
 exclude('\*\*/MainActivity$\*.class')  
}

3. In Terminal, type gradlew createJarDebug (you need to generate the obguile package, type gradlew createJarDebug proguard)

If prompted build success, the package succeeds

Jar package path: app/build/libs/test.jar

Copy to Unity engineer and tell them package name. Class name, interface name

2. call jar file

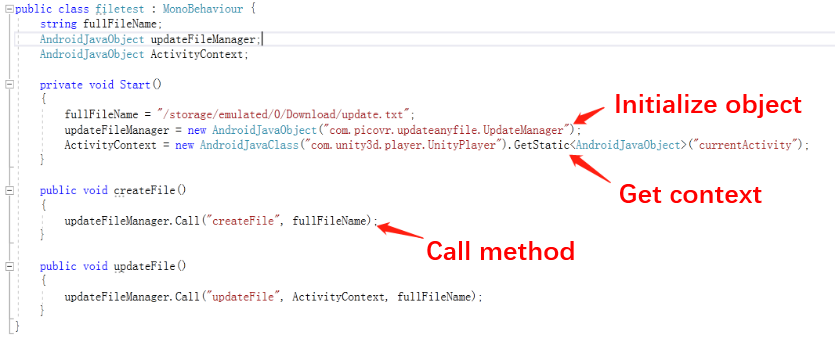
1. Export the Jar package and remember the package name of the Jar package. Class name, method name

2. Copy the Jar package to the Assets/Plugins/Android/ directory

3. write script

Using “packageName.ClassName” to initialize Object of Class in jar, as in picture below:

com.picovr.updateanyfile.UpdateManager

**

Method3: send Message using UnityPlayer.UnitySendMessage

1. write JAR
2. import classes.jar to app -> libs, right click it and choose Add As Library
3. inherit UnityPlayerActivity class
4. invoke UnityPlayer.UnitySendMessage(String object, String method, String arg) when needed.

eg: send Bluetooth state change state once u receive the broadcast.

… …

switch (message) {

case BluetoothAdapter.STATE\_OFF:

UnityPlayer.UnitySendMessage("SetState", "setBluetoothState", "bt\_state\_off");

break;

… …

1. remove the line setContentView(R.layout.activity\_main);
2. Add the following code in the app-> build.gradle root node (which is the same level as android and dependencies).

task createJarDebug(type: Jar,dependsOn: ['build']) {  
 baseName "test"  
 from 'build/intermediates/javac/debug/compileDebugJavaWithJavac/classe s'  
 include('\*\*')  
 exclude('\*\*/R.class')  
 exclude('\*\*/R$\*.class')

exclude('\*\*/BuildConfig.class')  
}

//需要混淆时添加以下代码

task proguard(type: proguard.gradle.ProGuardTask, dependsOn: buildJar) {  
// 输入路径  
 injars "build/libs/test.jar"  
// 输出路径  
 outjars "lib/testProguard.jar"  
// 添加配置信息  
 configuration 'proguard-rules.pro'  
}

1. In Terminal, type gradlew createJarDebug

You need to generate the conflation package by typing gradlew createJarDebug proguard

If prompted build success, the package succeeds

Jar package path: app/build/libs/test.jar

Copy to Unity engineer and tell them package name. Class name, interface name

1. **Invoke method**

1) Export the Jar package and remember the package name, class name and method name of the Jar package

2) Copy the Jar package to Assets/Plugins/Android/ path

3) Call the GameObject of s1 with the same name of UnitySendMessage(String s1, String s2, String s3) in the new Jar package, as "SetState" above.

4) When building a script, name it any other than "setBluetoothState", and create a method of the same name in the script whose parameter is a String data type.And mount the script on SetState.Once the UnitySendMessage in the jar is executed, the method is called and s3 is passed in as a parameter.

test.cs:

public class test : MonoBehaviour {

public Text wifiText;

public void setWifiState(string s)

{

wifiText.text = s;

}

## Attention

1. If there is a dependent jar package in the android project that is not in the Unity project, you need to copy it to the Unity project

2. Components registered in the android project manifest file, such as service, receiver, etc., should also copy the declared permissions into the unity project manifest