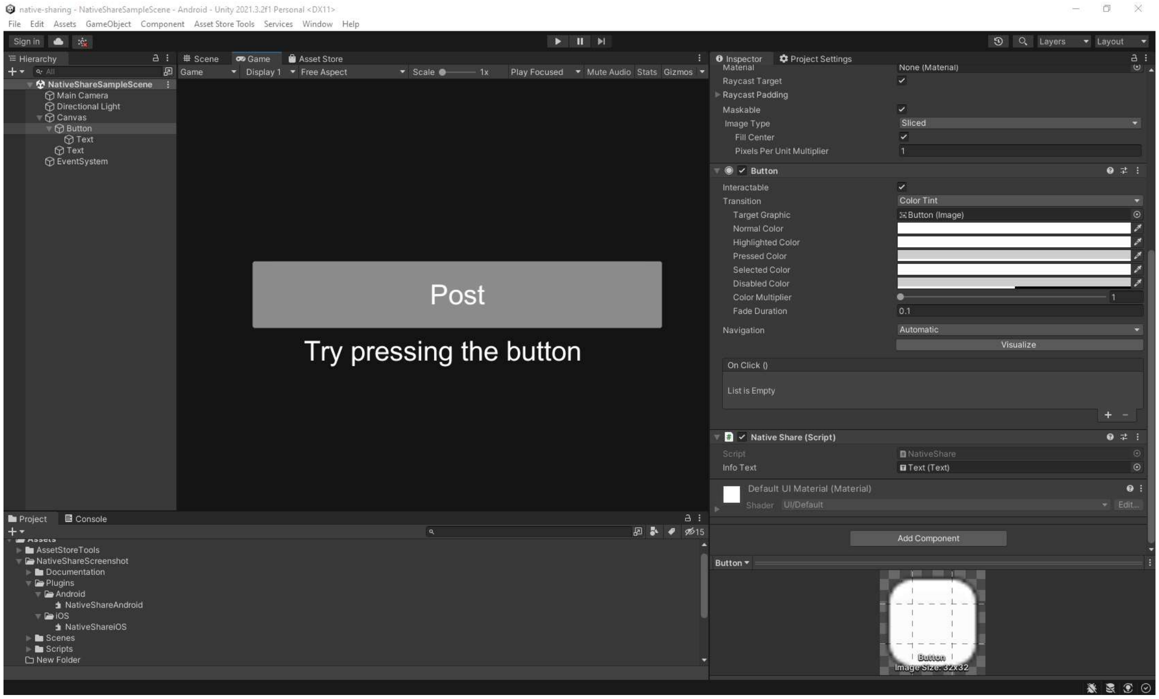
Native Share Screenshot



Material: None (Material)
Raycast Target: ☑
Raycast Padding
Maskable: ☑
Image Type: Sliced
Fill Center: ☑
Pixels Per Unit Multiplier: 1

☑ Button
Interactable: ☑
Transition: Color Tint
Target Graphic: Button (Image)
Normal Color
Highlighted Color
Pressed Color
Selected Color
Disabled Color
Color Multiplier: 1
Fade Duration: 0.1
Navigation: Automatic

On Click ()
List is Empty

☑ Native Share (Script)
Script: NativeShare
Info Text: Text (Text)

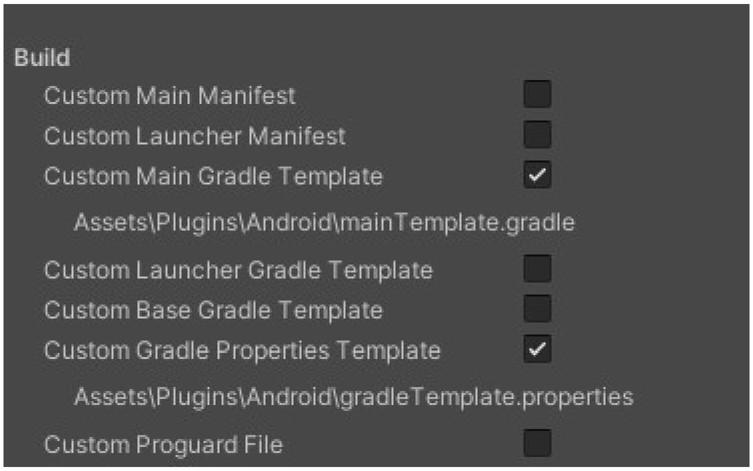
Default UI Material (Material)
Shader: UI/Default

Overview:

Use this asset to programmatically take a screenshot of your app and share it using using native share for iOS and Android. You can use this asset to share a screenshot of your high score to Facebook for example. It is simple and easy to use. Simply import the asset into your project and attach the script to your button...that's it! No additional coding or SDK setup is required. This works on Android 8+(using FileProvider),and on iOS 9+(using UIModalPresentationPopover). I have tried to make this as simple and easy to understand as possible, so you can get setup sharing screenshots of your game in a matter of minutes.

Quick Setup:

Import the asset into your project, and then simply drag the NativeShare.cs scrip onto the button of your choice that you would like to trigger sharing. For Android in Unity go to File → Build Settings → Player Settings → Publishing Settings → Under Build enable Custom Gradle Template.



Build

Custom Main Manifest ☐

Custom Launcher Manifest ☐

Custom Main Gradle Template ☑

Assets\Plugins\Android\mainTemplate.gradle

Custom Launcher Gradle Template ☐

Custom Base Gradle Template ☐

Custom Gradle Properties Template ☑

Assets\Plugins\Android\gradleTemplate.properties

Custom Proguard File ☐

You will need to modify two files here. You will see the path for our mainTemplate.gradle file, as well as the path for our gradleTemplate.properites file.

For the main.Template.gradle file, edit and append the following lines to the bottom of that file:

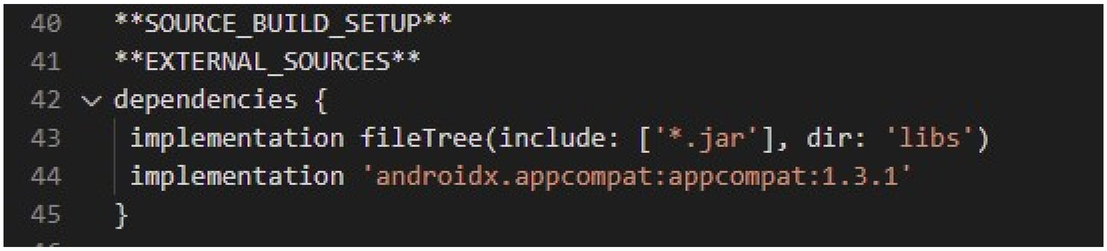
dependencies {

implementation fileTree(include: ['*.jar'], dir: 'libs')

implementation 'androidx.appcompat:appcompat:1.3.1'

}

So the end of the file should look like this:



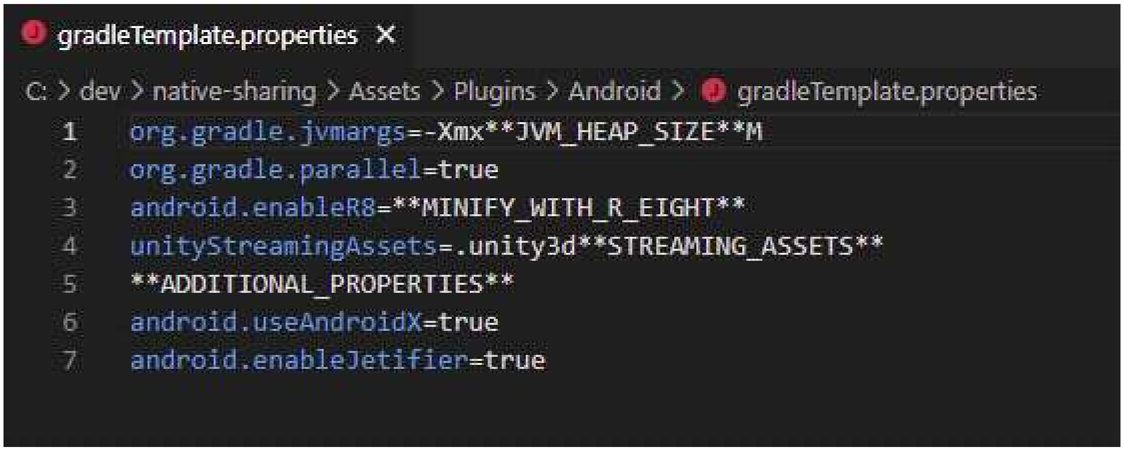
```
**SOURCE_BUILD_SETUP**
**EXTERNAL_SOURCES**
dependencies {
 implementation fileTree(include: ['*.jar'], dir: 'libs')
 implementation 'androidx.appcompat:appcompat:1.3.1'
}
```

The second file that you need to modify is the gradleTemplate.properties file. Append the last two lines to the bottom of that file:

android.useAndroidX=true

android.enableJetifier=true

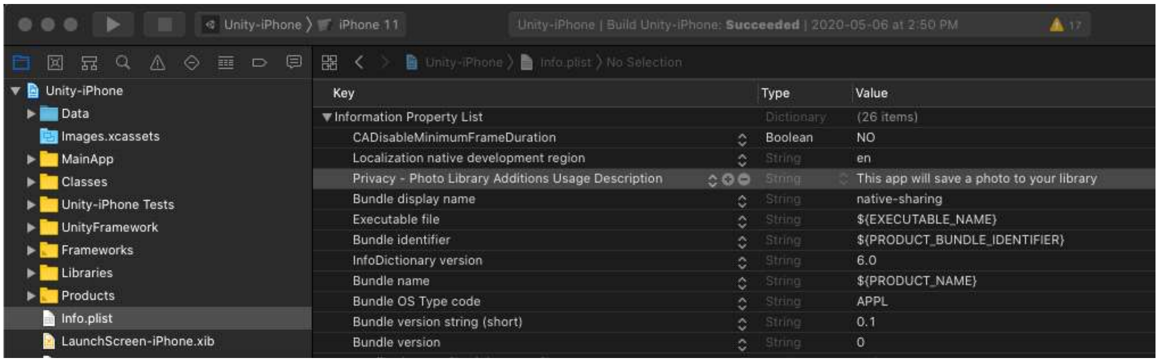
So it should look something like this:


gradleTemplate.properties
C: > dev > native-sharing > Assets > Plugins > Android > gradleTemplate.properties

```
org.gradle.jvmargs=-Xmx**JVM_HEAP_SIZE**M
org.gradle.parallel=true
android.enableR8=**MINIFY_WITH_R_EIGHT**
unityStreamingAssets=.unity3d**STREAMING_ASSETS**
**ADDITIONAL_PROPERTIES**
android.useAndroidX=true
android.enableJetifier=true
```

Setting up permissions for iOS:

If you are building for iOS, you will need to update the info.plist file in xcode to allow for the appropriate permissions. You can modify it in xcode by clicking on the info.plist file in the root of the file structure, and then under key click on the + icon and add Privacy - Photo Library Additions Usage Description, and add a key value.



Key	Type	Value
Information Property List	Dictionary	(26 items)
CADisableMinimumFrameDuration	Boolean	NO
Localization native development region	String	en
Privacy - Photo Library Additions Usage Description	String	This app will save a photo to your library
Bundle display name	String	native-sharing
Executable file	String	${EXECUTABLE_NAME}
Bundle identifier	String	${PRODUCT_BUNDLE_IDENTIFIER}
InfoDictionary version	String	6.0
Bundle name	String	${PRODUCT_NAME}
Bundle OS Type code	String	APPL
Bundle version string (short)	String	0.1
Bundle version	String	0

That should be it. Please note that the native sharing dialog will only appear when you are actually running on the actual Android or iOS device. It will not work in the editor. You shouldn’t need to do anything else. If you run into problems you can try building the sample scene and test to see if that is working.

Additional Configuration:

Info text assignment is optional. It is used in the sample scene just to give a bit more additional information on the screen. You can check out the sample scene to see it in use.