Android & iOS Native Screen Recorder

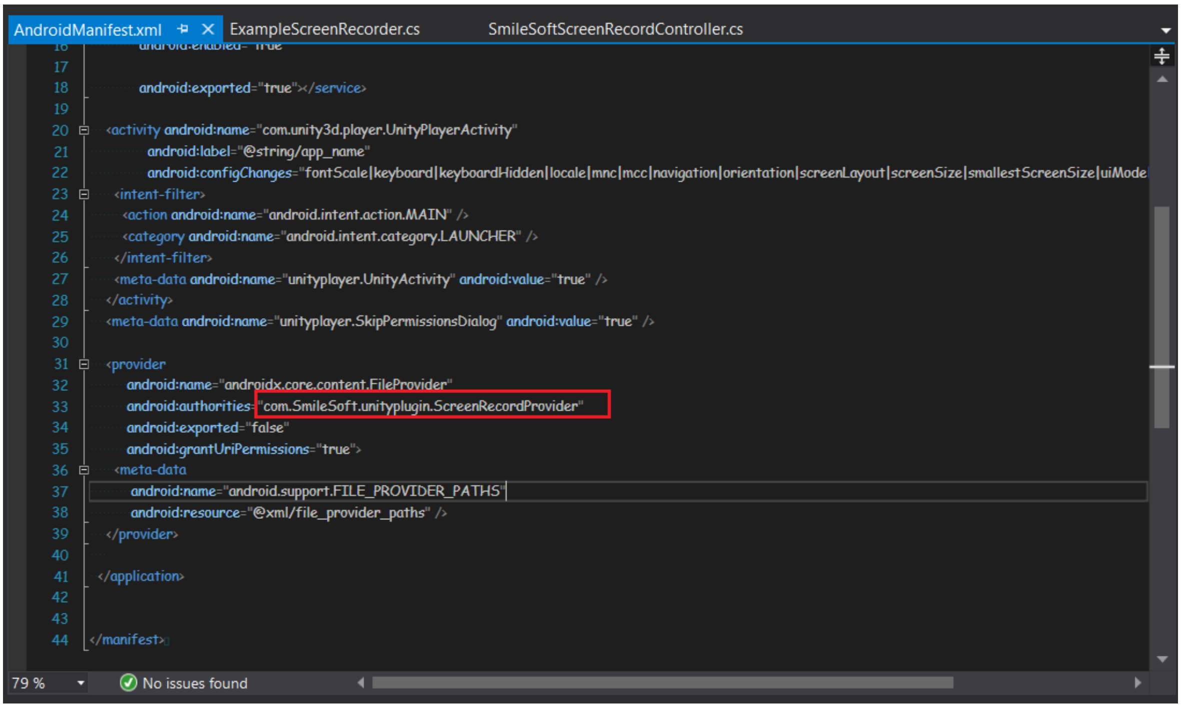
This plugin can record your gameplay without any issue. It uses android native **MediaRecorder** API and **Android Foreground Service** for record screen. So it will support from **Lollipop (5.1)** to newer versions even **Android 12(Latest One).**

In iOS it uses the iOS replayKit Library. It will support on all iPhone, iPad and iPod Touch devices with iOS 9.0 or above.

Instruction: After importing the project please drag **Screen Recorder** prefab in Scene Hierarchy from **Assets > SunShine Android Native Screen Recorder > Prefab > Screen Recorder**.

If you want to share the video using native share dialog then please do the following settings. **(Only Android)**

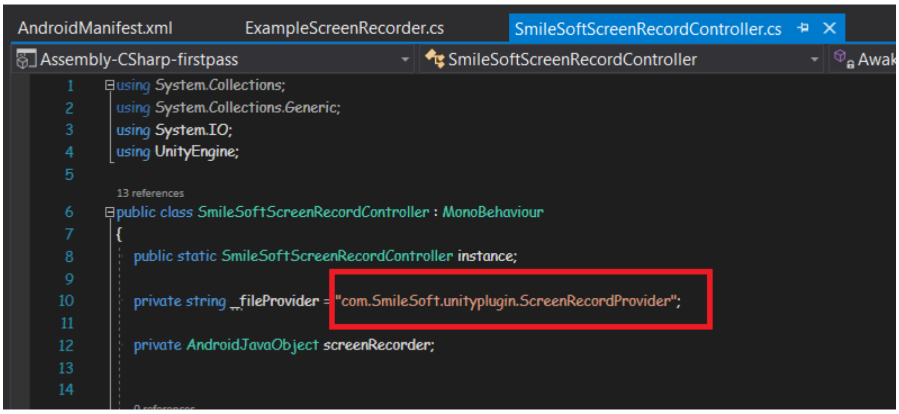
1. Open Android manifest file from **Assets > Plugins > Android > AndroidManifest.xml** . Now change the authority name with some unique one. We highly recommend you to use the package name.



```
          android:enabled="true"

          android:exported="true"></service>

    <activity android:name="com.unity3d.player.UnityPlayerActivity"
            android:label="@string/app_name"
            android:configChanges="fontScale|keyboard|keyboardHidden|locale|mnc|mcc|navigation|orientation|screenLayout|screenSize|smallestScreenSize|uiMode
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
      <meta-data android:name="unityplayer.UnityActivity" android:value="true" />
    </activity>
    <meta-data android:name="unityplayer.SkipPermissionsDialog" android:value="true" />

    <provider
        android:name="androidx.core.content.FileProvider"
        android:authorities="com.SmileSoft.unityplugin.ScreenRecordProvider"
        android:exported="false"
        android:grantUriPermissions="true">
      <meta-data
         android:name="android.support.FILE_PROVIDER_PATHS"
         android:resource="@xml/file_provider_paths" />
    </provider>

  </application>


</manifest>
```

79 % No issues found

2. Again Open **SmileSoftScreenRecordController.cs** from **Assets > SunShine Android Native Screen Recorder > Scripts > SmileSoftScreenRecordController.cs**. Now change the _fileProvider value which you set on the manifest file before.



AndroidManifest.xml
ExampleScreenRecorder.cs
SmileSoftScreenRecordController.cs
Assembly-CSharp-firstpass
SmileSoftScreenRecordController
Awak

```
using System.Collections;
using System.Collections.Generic;
using System.IO;
using UnityEngine;

13 references
public class SmileSoftScreenRecordController : MonoBehaviour
{
    public static SmileSoftScreenRecordController instance;

    private string _fileProvider = "com.SmileSoft.unityplugin.ScreenRecordProvider";

    private AndroidJavaObject screenRecorder;


```

You can also see this video For setting the file provider path value.

Now you are ready for calling the plugin API. The following functions are available in this plugin.

Functions:

1. **Start Record:** **SmileSoftScreenRecordController.instance.StartRecording()**

2. **Stop Record:** **SmileSoftScreenRecordController.instance.StopRecording()** . In android It returns the recorded file path. in iOS devices it will open the native preview window. From there you can edit, discard or save the recorded video in camera roll even you can share it instantly with your sharing apps.

3. **Record Audio:** **SmileSoftScreenRecordController.instance.SetAudioCapabilities(isAudioRecording)**. Where **isAudioRecording** is a **Boolean** type parameter. This will record audio from the device mic.

4. **Set File Destination:** **SmileSoftScreenRecordController.instance.SetVideoDestination(destination)**. Where **destination** is a **String** type value. By default the destination is **External Storage** directory. **(Only Android)**

5. **Set File Name:** **SmileSoftScreenRecordController.instance.SetVideoName(fileName)**. Where **fileName** is a **String** type value. **(Only Android)**

6. **Folder Name:** You can create a new folder and save your video there. Just call **SmileSoftScreenRecordController.instance.SetStoredFolderName(folderName)** function. Here **folderName** is a **String** type value. **(Only Android)**

7. **Set Gallery Add Capabilities:** **SmileSoftScreenRecordController.instance.SetGalleryAddingCapabilities(canAddIntoGallery)**. Where **canAddintoGallery** is a **boolean** type value.If it is true then video file will be added into device gallery.By default this value is **true.** **(Only Android)**

8. **Bit Rate:** **SmileSoftScreenRecordController.instance.SetBitRate(bitrate)**. **bitrate** is an **integer** type parameter. Higher bitrate means high quality video. **(Only Android)**

9. **Preview:** If you want to see the preview in the native video view then just call this function **SmileSoftScreenRecordController.instance.PreviewVideo(_recordedFilePath)**.Here **recordedFilePath** is an **string** type parameter. For iOS you do not have to pass the filepath.

10. **Share Video:** You can also Share the video using native share dialog. Just call **SmileSoftScreenRecordController.instance.ShareVideo(_recordedFilePath, message, share title)**. It takes

three parameters and all of them are **strings**. First one is the **filePath**, second one is the **share message** and third one is **share title**. **(Only Android)**

11. **Video Size:** **SmileSoftScreenRecordController.instance.SetVideoSize(width,height)**. Both **width** and **height** are integer type parameters. **(Only Android)**

12. **Video Encoder:** **SmileSoftScreenRecordController.instance.SetVideoEncoder(videoEncoder)**. Here **videoEncoder** is an integer type parameter. Before setting this please look at the Android Developer official site for supporting encoders in different Android API versions. **(Only Android)**

    To get a clear idea please see the example scene from **Assets > SunShine Android Native Screen Recorder > Example > Example Scene.**

**** Here 3-9 number functions are optional. You can use all of them or none. But please be sure that these (3-9) functions are called before calling the Start Record function. ****

CallBack Functions : You get the following callbacks in **SmileSoftScreenRecordController.cs script.** **(Only Android)**

OnRecordPermissionGranted - It will call when Record permission popup opens and users accept / reject the permission.
Parameter : Status (string)
True - Accept permission.
False - Declined Permission

OnRecordStartStatus - It will call when a user tries to record.
Parameter : Status (string)
True - Record Start Successfully
False - Record Failed