Installation of Crittercism

for Unity Android

Table of Contents

Relevant Package Content

Plugins/Android
Plugins/Crittercism_Android_Scripts
Unity_Android_Integration
TestAndroid

Steps for Basic Integration

In the documentation below, **{Unity Project}** refers to your project path, or the test application included in this package.

Step 1 : Decompress

Decompress the *Crittercism_Unity* package. This will contain a copy of documentation, the plugins for Android and iOS, and simple integration test apps for both iOS and Android.

Step 2 : Drag and Drop

Drag and drop the following items to their appropriate counterparts in the Unity Project:

Android_Unity/Plugins/Android   --->   **Assets/Plugins/Android**
Android_Unity/Plugins/Crittercism_Android_Scripts   --->   **Assets/Plugins/Crittercism_Android_Scripts**

If you already have other Android Plugins, the contents of that folder should be copied over.

Step 3 : Android Manifest

A custom **AndroidManifiest.xml** must be added to the **{Unity Project}**/Plugins/Android folder. Inside the file, the following permissions must be added for communication:

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```
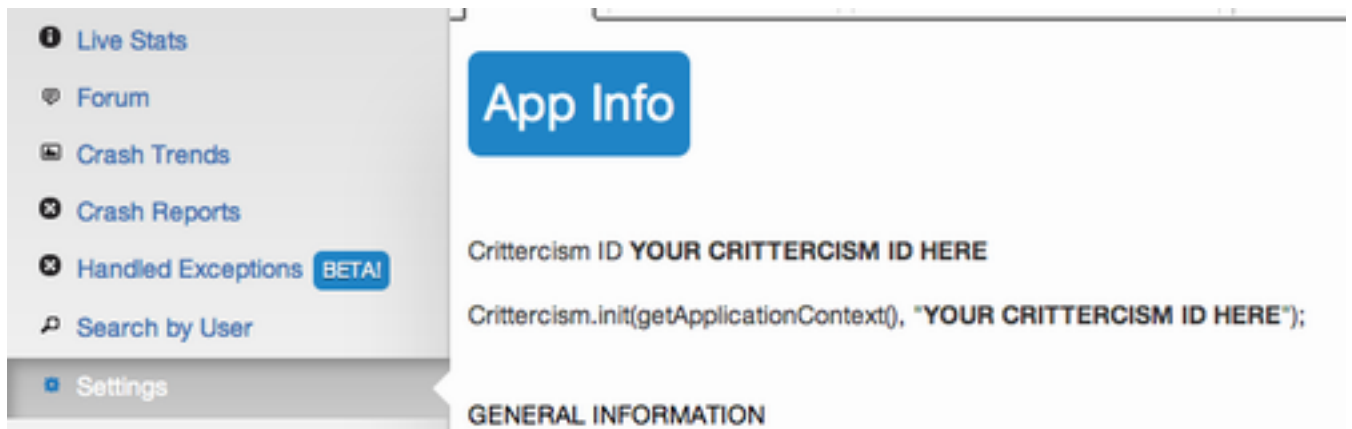
Don't forget, you can look to the Android Test App for an example of how this is done!

Step 4 : Connecting with your Crittercism ID

Open the strings.xml in (Unity Project)/Plugins/Android/res/values/. You will need to change the following line:

```
<string name="CrittercismAppID">/*Your Crittercism ID Here*/</string>
```

Replace **/*Your Key Here*/** with the key from the Crittercism Web Portal, as seen below:

Step 5 : Crittercism Initialization

In Unity, in the application launching scene Drag the **Crittercism_Android_Scripts/CrittercismInit.cs** script onto the Camera or a GameObject. The **CrittercismInit.cs** will start Crittercism debugging in Unity3d at scene load. (See Crittercism at App Launch.) This script will destroy itself in newer versions of Unity3d at runtime. In versions before Unity 3.4.2, it will stay active in the scene until an Update call.

You can look in the Inspector of the game object you've added the script to confirm the script is now associated with the object, like the example below:
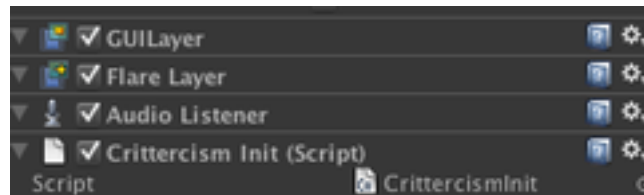
Step 6 : Build and Run!

You've finished the basic integration! Now you can to Build and Run with an appropriate testing physical or virtual device.

Optional Integrations

Crittercism at App Launch

This will provide debugging of Unity3d crashes. This will normally require overriding the Unity3d Activity or Application. For convenience a custom Application is provided in the **CrittercismAndroid.jar**.

A custom **AndroidManifiest.xml** must be included in the **{Unity Project}**/Plugins/Android directory. The following line must be added to the **AndroidManifiest** in the <application> tag:

```
android:name="com.Crittercism.UnityPlugin.ExtendedApplication"
```

Making Crittercism Init Highest Priority

Select Edit->Project Settings->Script Execution Order, and add the **CrittercismsInit.cs** script to the list with the highest priority . This will prevent any other files from executing before Crittercism for Unity is enabled. If crashes occur before Crittercism is started in Unity3d, they will not be captured and sent to the web portal.

Test GUI within Your App

In Unity, drag **Plugins/Crittercism_Android_Scripts/CrittercismTestGUI.cs** file to the camera. This provides a test interface for Unity3d that will throw exceptions.

Change Log

07-10-2012   Removed the activities used for support forums in Step 3.

12-13-2012   Updated file structure for new structure and related mentions in doc.