Gallery Screenshot

iOS / Android / Windows Phone 8 Plugin

This Plugin will take a screenshot, and register the image file so that it appears in your media library (camera roll) on iOS, Android & Windows Phone 8.

In addition to saving screenshots into the media library, this Plugin can also save an existing image, e.g. a jpeg loaded as a Texture2D. Please read this guide and take a look at the included Demo Scene to see how to do this (Demo scene should be ran at portrait res, e.g. 720x1280).

There is a full C# scripting reference here: http://secondfury.com/galleryscreenshot/

Android Setup

The asset package should import to *Assets/Plugins/* - please ensure this is the location of the files otherwise you will run into errors.

If you are using Android, you will need to go into **Build Settings** and access the **Player Settings** panel. Change **Write Access** to **External (SDCard)** to allow the screenshots to save.
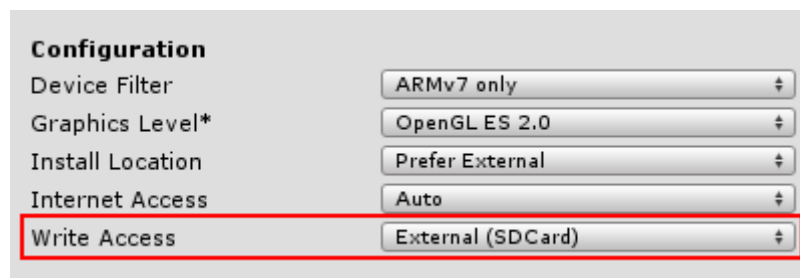
Windows Phone 8 Setup

Please first make sure your **Platform** is set to **Windows Phone 8** in **Build Settings**. You must also ensure the capability **ID_CAP_MEDIALIB_PHOTO** is specified in the app manifest. When you build your app this will be **<BUILDPATH>/Properties/WMAppManifest.xml**.

<Capability Name="ID_CAP_MEDIALIB_PHOTO"/>

An extra step for **Unity 5** users is the .dll files must have their import settings set in the Inspector.

Assets/Plugins/WP8Screenshot.dll
Platforms: All platforms checked except WP8Player

Assets/Plugins/WP8/WP8Screenshot.dll
Platforms: Check WP8Player only
Placeholder: Assets/Plugins/WP8Screenshot.dll

iOS Setup

There are no additional steps.

Saving a Screenshot

ScreenshotManager.SaveScreenshot(string fileName, string albumName = "MyScreenshots",
string fileType = "jpeg", Rect screenArea = default(Rect))

The code for taking a Screenshot has been updated since previous versions of the plugin. To take and save a screenshot in its simplest form, write the following code:

ScreenshotManager.SaveScreenshot ("ScreenshotName", "AlbumName");

Where **ScreenshotName** is the file name you would like the screenshot to be saved as, and **AlbumName** is the folder that the screenshot will be put into on compatible devices (this will appear as an album name in your gallery on Android devices). A time stamp will be automatically appended to each image to prevent overwrites, in the form MyScreenshot_hh-mm-ss_dd-MM-yy. This can easily be modified in the ScreenshotManager.cs source if you wish.

N.B. On older devices (iPhone 3 / Galaxy S) it can take a couple of seconds to write the picture file to the SD card. If the user exits before this finishes, the image won't be registered to the gallery. Therefore it is recommended to have a "saving image" graphic on screen for these older devices.

Events

public static event Action<Texture2D> OnScreenshotTaken;

Fires when a screenshot has finished taking, and sends back the screenshot as a Texture2D.

public static event Action<string> OnScreenshotSaved;

Fires when a screenshot has finished saving to disk, and passes back its file location as a string.

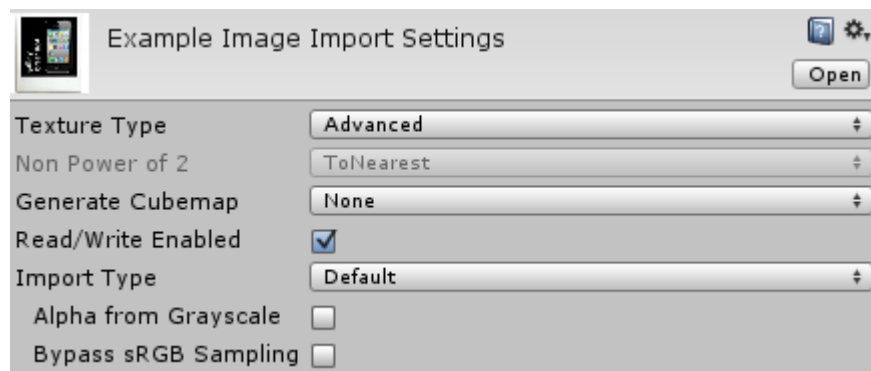Please see the Demo Scene for examples of using these Events, and remember there is a full C# scripting reference here: http://secondfury.com/galleryscreenshot/

Using the Plugin to Save an Existing Image

ScreenshotManager.SaveImage(Texture2D texture, string fileName, string fileType = "jpeg")

So in its simplest form, to save a existing Texture2D named tex to disk as "MyImage.jpeg":

ScreenshotManager.SaveImage(tex, "MyImage");

Remember you must ensure your source texture is set to **Read/Write Enabled** in the Unity editor Import Settings for this to work.

Events

public static event Action<string> OnImageSaved;

Fires when an image has finished saving to disk, and passes back its file location as a string.

Please see the Demo Scene for examples of using these Events, and remember there is a full C# scripting reference here: http://secondfury.com/galleryscreenshot/

Finally thanks for purchasing this plugin and I hope that it helps you with your project!