

# Gallery Screenshot

## iOS / Android / Windows Phone 8 Plugin

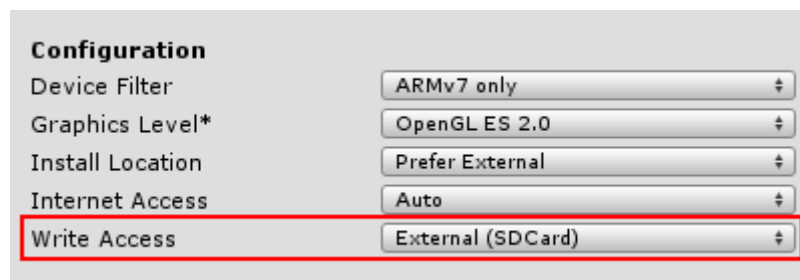
This Plugin will take a screenshot, and register the image file so that it appears in your media library (camera roll) on iOS, Android & Windows Phone 8. Unity 4.2 or greater is required to use the Windows Phone 8 support. Android and iOS has been tested on 3.5.7 onwards.

As well as putting screenshots to the media library, the Plugin can also take an existing image (like a Texture2D) and put it in the media library. Please read this guide and take a look at the included Demo Scene to see how to do this.

## 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

For WP8, extract the files in WP8\_DLL.zip to the **Plugins** folder so that the two copies of WP8Screenshot.dll and WP8Screenshot.pdb remain in their respective directories e.g.

**Assets/Plugins/WP8Screenshot.dll**

**Assets/Plugins/WP8/WP8Screenshot.dll**

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/WMAAppManifest.xml.

## iOS Setup

There are no additional steps.

## Saving a Screenshot

The Plugin is initialised through a Coroutine. To take a screenshot write the following code:

```
StartCoroutine(ScreenshotManager.Save( "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 number and time stamp will be automatically appended to each image to prevent overwrites, i.e. MyScreenshot\_1\_12-03-13. This can easily be modified in the ScreenshotManager.cs source if you wish.

**There is a third parameter – a boolean which specifies if you would like a call back when saving completes (defaults to false), if you are unsure on how to use call backs please see the included Demo Scene for an example. This call back also returns the location of the saved screenshot.**

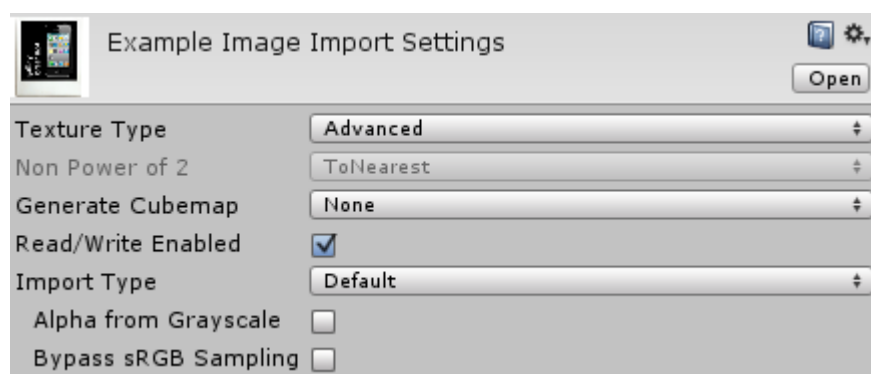
N.B. On older devices (iPhone 3 / Galaxy S) it can take a couple of seconds to write the picture file to the SD card. If the user exits before this finishes, the photo won't be registered to the gallery. Therefore it is recommended to have a "saving photo" graphic on screen for these older devices.

## Using the Plugin to Save an Existing Image

Once again this is initialised through a Coroutine. To register the image to your gallery write:

```
StartCoroutine(ScreenshotManager.SaveExisting( byte[], "FileName" ));
```

Where ***byte[]*** is a PNG encoded byte array of your texture and ***FilePath*** is your desired file name. Remember you must ensure your source texture is set to **Read/Write Enabled** in the Unity editor Import Settings for this to work.



Again there is an example provided as part of the Demo Scene (which also demonstrates use of the call back). I hope you find this Plugin useful and good luck with your project!