



Mobile Game and App Development the Easy Way

Developed and maintained by Pocketeers Limited (<http://www.pocketeers.co.uk>). For support please visit <http://www.appeasymobile.com>

This document is protected under copyright to Pocketeers Limited @2012.

Version 1.4

Table of Contents

What is the AppEasy Project Manager?.....	
Installing AppEasy Project Manager.....	
Using AppEasy Project Manager.....	
Tools Panel.....	
Assets Panel.....	
App Configuration Panel.....	
Output Panel.....	
Android Setup, Signing and App Store Submissions.....	
App Store Submission.....	
iOS Setup, Signing and App Store Submissions.....	
QA.....	
The Future.....	
Support.....	

What is the AppEasy Project Manager?

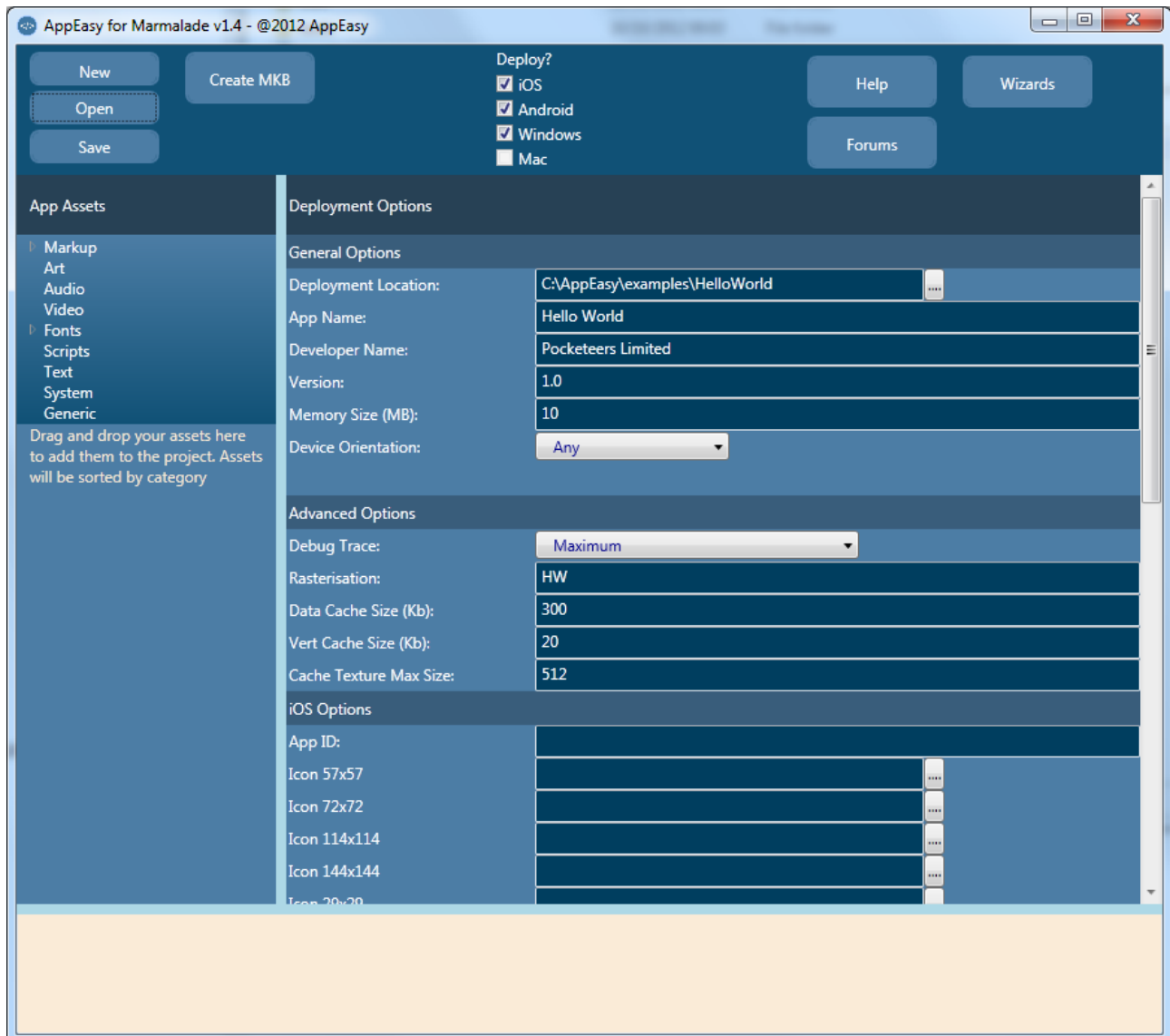
The AppEasy Project Manager is a Windows based tool that allows you to collect all of your AppEasy / IwGame app and game assets together and set common Marmalade deployment and project settings then generate a fully working Marmalade MKB project that can be built and deployed to any of the supported Marmalade platforms. AppEasy project manager provides a convenient drag and drop interface that allows you to drag and drop splash screens, icons, provisioning profiles and assets etc into the project and AppEasy takes care of exporting the correct MKB and ICF file settings.

AppEasy is a combination of the IwGame Engine and its powerful XOML mark-up language, Lua and a desktop project manager for managing assets and setting up common project settings. AppEasy for Marmalade enables rapid development of games and apps without the need to learn the intricacies of the Marmalade SDK or even the need to use C or C++. Its perfectly viable to create apps purely in XOML mark-up or a combination of XOML and Lua.

Installing AppEasy Project Manager

To install AppEasy visit <http://www.appeasymobile.com> and download AppEasy for Marmalade and unzip to your C drive. **Note that as AppEasy for Marmalade is still in development the project manager is currently locked to the C drive.** This will be changed in the future so that it can be installed to any drive / folder.

Using AppEasy Project Manager



The AppEasy Project Manager is split into 4 sections. The upper panel that contains a collection of buttons called the Tools Panel. The left hand panel which contains the App Assets list is called the Assets Panel and contains a list of all assets that are in the current project. The large middle panel is know as the App Configuration Panel and is where all application info is shown. The bottom white panel is the Output Panel which shows errors, messages and warnings that are generated whilst AppEasy is carrying out various tasks

Tools Panel

The Tools Panel is where you drag and drop an AppEasy project file to open it. It also hosts a variety of buttons and other controls to allow you to test and deploy your apps. Lets take a look at those controls in detail:

- New – Creates a new project, removing the previous project from the project manager
- Open – Opens an existing AppEasy project for modification, testing and deployment
- Save – Saves the current AppEasy project
- Create MKB – Creates a Marmalade project and copies your assets into it.
- Deploy? - This section allows you to deploy MKB and App.icf settings for each platform type. If a section is unchecked then those settings will not be exported when the project is created.
- Help – Takes you to the AppEasy web site
- Forum – Takes you to the AppEasy forums
- Wizards – Launches the AppEasy easy wizards which can be used to help walk you through a whole host of processes including iOS and Android account creation, device provisioning, certificate generation, App store submissions and more..

You can also open an AppEasy project by dragging and dropping the project file onto the Tools Panel.

Assets Panel

The Assets Panel is a drag and drop enabled panel that organises the projects assets into asset types including:

- Markup – Contains mark-up files
- Artwork – Contains art work files
- Audio – Contains audio files
- Video – Contains video files
- Fonts – Contains true type font files
- Scripts – Contains script files
- Text – Contains text files

- System – Contains system files
- Generic – Contains files that do not fit into any other section

To Add assets to AppEasy simply select them in explorer and and drag and drop them onto the Assets Panel. All dropped assets will be organised into their correct asset categories.

You can expand / contract asset categories to show / hide their contents. You can also navigate the resources using the cursor keys. Double clicking an asset or pressing the enter key on an asset will open it using the systems designated viewer / editor.

Pressing the Delete key on an asset will delete it from the project but will not remove it from disk.

App Configuration Panel

The App Configuration Panel contains information about your app that is used to add icons, splash screens, decide its name etc. Dragging and dropping files onto this section will organise them into the correct places. For example dropping a selection of different sized icons onto the configuration panel will automatically place the correct sized icons into the correct places. Application configuration options include:

General Options

- Deployment Location – This is the drive:folder where your application will be deployed to
- App Name – The name of your applications
- Developer Name: Your developer name as set in the App stores
- Version – The version number of your application (1.0 for example)
- Memory Size – This is an estimate as to how much memory your application can use at one time in number of megabytes (default is 20)
- Device Orientation – You can use this option to fix the orientation of your deployed application (only works on device). Default is set to “any” which tells the app that it is allowed to switch orientations when the device changes orientation

Advanced Options

- Debug Trace – This option allows you to specify how much debug trace information will appear in the debug output.
- Rasterisation (advanced) – This option allows you to change the renderer used by the device to draw your app. Possible values include HW (default), SW, SW/HW and HW/SW. Choosing HW will use hardware rendering, SW will use software rendering. SW/HW will start in software but switch to hardware if

- supported. HW/SW will start in hardware but switch to software if supported
- Data Cache Size (advanced): This is the number of kilobyte that are reserved for the data cache. Default is 200, if you draw too many actors or text glyphs at the same time then you may need to increase this
 - Vert Cache Size (advanced): – This is the number of thousands of vertices that can be rendered in one go. Default is 20, if you draw too many actors or text glyphs at the same time then you may need to increase this. If you are getting unexpected crashes then try gradually increasing this value
 - Cache Texture Size (advanced): This is the maximum texture size that will be used to cache font glyphs. Default is 512, if you render large fonts and characters are missing then you should increase this size

iOS Options

- App ID: This is the App ID for your app as supplied by Apple's provisioning portal (for example. com.companyname.gamename). This is only required for iOS app store deployment
- Icon57x57 – 57x57 pixel PNG image file that is used to display your applications icon on none retina iPhone displays. This image should not contain an alpha channel
- Icon72x72 – 72x72 pixel PNG image file that is used to display your applications icon on none retina iPad displays. This image should not contain an alpha channel
- Icon114x114 – 114x114 pixel PNG image file that is used to display your applications icon on retina iPhone displays (iPhone4+). This image should not contain an alpha channel
- Icon144x144 – 144x144 pixel PNG image file that is used to display your applications icon on retina iPad displays (iPad 3). This image should not contain an alpha channel
- Icon29x29 – 29x29 pixel PNG image file that is displayed in iPhone settings and Spotlight. This image should not contain an alpha channel
- Icon58x58 – 58x58 pixel PNG image file that is displayed in iPhone retina settings and Spotlight. This image should not contain an alpha channel
- Icon50x50 – 50x50 pixel PNG image file that is displayed in iPad Spotlight. This image should not contain an alpha channel
- Icon100x100 – 100x100 pixel PNG image file that is displayed in iPad3 Spotlight. This image should not contain an alpha channel

- Splash iPhone – PNG image file that is used to display a 480x320 or 320x480 splash screen on boot on none retina iPhone displays. This image should not contain an alpha channel
- Splash iPhone (retina) – PNG image file that is used to display a 960x640 or 640x960 splash screen on boot on retina iPhone displays. This image should not contain an alpha channel
- Splash iPad – PNG image file that is used to display a 1024x768 or 768x1024 splash screen on boot on none retina iPad displays. This image should not contain an alpha channel
- Splash iPad retina – PNG image file that is used to display a 2048x1536 or 1536x2048 splash screen on boot on retina iPad displays. This image should not contain an alpha channel
- Distribution Provisioning Profile – This file is retrieved from Apples provisioning portal and will be called something like appname.mobileprovision. This file is only required when creating a build for the Apple App Store. See the AppEasy Wizard for information on how to create this file. You cannot deploy apps to the Apple app store without this file.
- info.plist (advanced) – This allows you to supply your own info.plist (optional)

Android Options

- Package Name: This is the package name for your app as it will appear on Google Play (In com.companyname.appname format). Note that appname must match the name of the app (in lower case and excluding any spaces). For example if the application is called Actor Example then appname should be actorexample
- Icon36x36 – 36x36 pixel PNG image file that is used to display your application icon on low resolution Android devices
- Icon48x48 – 48x48 pixel PNG image file that is used to display your application icon on medium resolution Android devices
- Icon72x72 – 72x72 pixel PNG image file that is used to display your application icon on high resolution Android devices
- Splash Screen – A JPEG file that will be used to display a splash screen on all Android devices
- Splash Screen Width – The width of the splash screen (used for image centring)
- Splash Screen Height – The height of the splash screen (used for image

centring)

- AndroidManifest – This allows you to specify your own custom AndroidManifest file (optional)
- Keystore Name – The name of your Android signing key store (required for store deployment)
- Keystore Password - The password used to access your keystore (required for store deployment)
- Keystore Alias - The alias used to by your keystore (required for store deployment)

Windows Options

- Window Size – Size of window to deploy Windows store build
- Full Screen – Enable if you would like your Windows build to run in full screen mode

Output Panel

The output panel displays info, warnings and errors that occur during AppEasy's operation.

Android Setup, Signing and App Store Submissions

Please refer to the Android Wizard for step by step instructions showing how to do the following:

- Create an Android developer account
- Create key store
- Install to device
- Submitting apps to the app store

The Android wizard is currently unavailable.

App Store Submission

You have many options when it comes to publishing your Android apps. Here are a few of the many Android app stores that are currently available to sell your wares:

- Google Android Market – <http://developer.android.com>
- Appia – <http://www.appia.com>
- Slideme – <http://slideme.org>
- Mobango - <http://www.mobango.com>
- Samsung Apps – <http://seller.samsungapps.com>
- Amazon App Store – <http://developer.amazon.com>
- Vodafone – <http://widget.vodafone.com>
- Verizon Wireless – <http://developer.verizon.com>
- Cellmania – http://www.cellmania.com/content_providers
- Mobihand / Mobireach – <http://www.mobireach.com>
- MiKandi – <http://developer.mikandi.com>
- LG World – <http://developer.lgmobile.com>
- GetJar - <http://developer.getjar.com>
- Motorola shop4apps – <http://developer.motorola.com>
- Handster – <http://www.handster.com/developers.php>
- Optus - <http://partnerconnect.optusnet.com.au>
- Butterscotch / Tucows – <http://www.butterscotch.com>
- AppsLib – <http://www.appslib.com>
- Fasmicro – <http://apps.fasmicro.net/developers>
- Pdassi – <http://pdassi.de>
- Android Pit – <http://www.androidpit.com>

AppEasy Project Manager Guide – Making Apps Just Got Easy

- Camangi Market – <http://www.camangimarket.com>
- DoComo Market – <http://www.docomo-market.info>
- Sony Ericsson e-store – <http://estore.ericsson.com>
- FastApp Store - <http://www.fastapp.com>
- M1 App Store – <http://appstore.m1.com.sg/developer>
- MobileApps – <http://www.mobileapps.com/mobileapps/main/developer.jsp>
- Phoload – <http://www.phoload.com/developer>
- MV Store – <http://www.mvstore.vn>
- Exent – <http://www.exent.com>
- NeXva – <http://corp.nexva.com/content-providers>
- 1Mobile – <http://author.1mobile.com>
- AndroidFreeware - <http://www.androidfreeware.net>
- Mobiles24 – <http://www.mobiles24.com>
- CNET – <http://upload.cnet.com>
- Soc.io Mall – <http://soc.io>
- AppBackr – <http://www.appbackr.com>
- AndroidApps - <http://www.androidapps.com>
- AndroidTapp - <http://www.androidtapp.com>
- GiveMeApps - <http://www.givemeapps.com>
- Nook Store - <http://nookdeveloper.barnesandnoble.com>
- Android Games - <http://android-games.org>

You should try to get your products out on at least the following markets:

- Google Android Market – <http://developer.android.com>
- Appia – <http://www.appia.com>
- Slideme – <http://slideme.org>
- Samsung Apps – <http://seller.samsungapps.com>
- Amazon App Store – <http://developer.amazon.com>
- GetJar - <http://developer.getjar.com>
- Nook Store - <http://nookdeveloper.barnesandnoble.com>

iOS Setup, Signing and App Store Submissions

Please refer to the Apple Wizard for step by step instructions showing how to do the following:

- Create an iOS developer account
- Create signing certificate request
- Download and install developer and distribution certificates
- Add test devices
- Provision devices for testing
- Submitting apps to the app store

To launch the Apple Wizard click the "Wizards" button in the AppEasy Project Manager

QA

Q) How and why do I need to remove alpha channels from PNG's for icons?

A) Apples App Store does not like PNG's that contain an alpha channel. To remove this channel yuo can use an art editing tool called GIMP. Which can be downloaded for free from <http://www.gimp.org/> . To remove the alpha channel from a PNG file, open the file in GIMP and select Image->Flatten Image then resave the image.

Q) How long does it take to get my iOS app approved by Apple?

A) Depends how busy Apple are, usually 10-14 days is about right. Updated app approval is usually much shorter.

Q) How long does it take to get my Android app approved?

A) This depends entirely on the store. Google Play like many other Android app stores have no approval process, you can simply upload your app and start selling it within a few hours. Amazon App Store does test your app however and will take around 2 weeks to approve you app.

Q) Where can I sell Windows apps and games created with AppEasy?

A) You can sell them anywhere on the internet, even from yiour ownm web site. Some more notable services include Steam (<http://store.steampowered.com/>) and GamersGate (<http://www.gamersgate.co.uk/>)

The Future

Future additions to AppEasy for Marmalade include:

- Android and BlackBerry OS wizards
- Addition of deployment settings for BlackBerry OS and other platforms
- Integrate Marmalade compile process and simulator testing

Support

Please visit our support forums at <http://www.appeasymobile.com/forum/>