Marmalade SDK FAQ v1.02

Produced and maintained by Mat Hopwood – http://www.drmop.com

For the latest version of thsi FAQ please check out http://www.drmop.com

Welcome to the Marmalade SDK developers FAQ. Here you can find solutions to problems that are encountered and resolved by other Marmalade developers:

Table of Contents

General

Q) What is the Marmalade SDK?

The Marmalade SDK is a powerful mature and battle tested cross platform development SDK that allows the developer to write native code once and run it across a multitude of different platforms using either C/C++ or HTML5 / JavaScript. The Marmalade SDK also comes with x86 and ARM simulators to easily test code as well as various tools to facilitate asset conversion, font generation etc.

Q) Is the Marmalade SDK free to use?

A) No, although there is a free 90 day trial that you can use to test out the SDK. Licensing starts from as little as $149 per year. For full licensing information see
http://www.madewithmarmalade.com/buy

Q) Are there any well known / famous games or apps that were created with the Marmalade SDK?

A) Yes there are quite a few including Draw Something, Backbreaker series and Pro Evolution Soccer

Q) Can I really write code once and deploy it to so many platforms?

A) Absolutely

Licensing

Q) How much does it cost to use the Marmalade SDK?

A) There is a free 90 day trial that you can use to test out the SDK. Licensing starts from as little as $149. For full licensing information see http://www.madewithmarmalade.com/buy

Q) I installed the Marmalade SDK to a 2nd computer but I cannot activate my license, why?

A) You can only have a single license active on one computer at one time. To use the Marmalade SDK on a different computer you should firstly go to the license management section of your account on www.madewithmarmalade.com and deactivate the license. You can now activate the license on the new computer.

Q) Can I sell an app that I created with the Marmalade SDK free trial?

A) No, it is for test purposes only

Getting Started

Q) I'm new to the Marmalade SDK, how do I get started?

A) The Marmalade SDK comes with a wealth of examples that you can check out in the Examples folder of the Marmalade installation. DrMop also keeps a collection of his own tutorials at http://www.drmop.com/index.php/marmalade-sdk-tutorials/ which are a great for walking new Marmalade developers through the basics.

Q) What is the app.icf file:?

A) The application configuration file (app.icf usually located in the projects data folder) is used to set various settings that Marmalade can read and act upon at run-time. The file allows you to specify many different options such as how much memory to allocate to your program, which device orientations your app can use etc.

Q) What is a MKB file?

A) A Marmalade Project file (MKB file) is a file that you create that describes your project to Marmalade. Once created you open the file and Marmalade will generate a Visual Studio solution on Windows or an XCode project file on Mac that you can use to build your project. The MKB file contains a number of important sections including:
- files – specifies which files you would like to include in your build
- assets – specifies which assets you would like to include in your build
- subprojects – specifies other projects to include in your project (kind of like libraries with source included)
- deployments – specifies common and per platform parameters that are used when deploying your app to the various platforms

Q) What is a MKF file?

A) A Marmalade Sub Project file (MKF file) is a file that you can create to contain a project that you want to include into your normal build but include it as a kind of library of code. This is useful for abstracting away the engine portions of your game or app into a separate re-usable project as well as adding in other modules written by other developers. Many of the Marmalade sub systems are implemented as sub projects to help cut down on the amount of code included in a project. For example, there is no use including the code that supports the Facebook API or the Resource Manager API if you are not going to be using them in your app. Note that including some sub projects will automatically include other sub projects behind the scenes as they have a dependency upon them.

General Development

Q) Can I use the Marmalade SDK on Windows?

A) Yes using Microsoft Visual Studio, including the free express versions

Q) Can I use the Marmalade SDK on Mac OS?

A) Yes, using XCode

Q) Can I use the Marmalade SDK on Linux?

A) No, not at this time

Q) I compiled my app but the build failed and the IDE asked me if I would like to reload the project. I did so but all of my open windows disappeared. How come?

A) This happens when you make a change to the projects MKB file whilst the project was open in the IDE. When you hit build, Marmalade detected that the MKB file had changed, hence the project has also changed (for example, addition of new files, removal of old files). At this point, Marmalade cannot continue the build as files may have been added or removed. The only solution is to halt the build and reload the project in the IDE. If you are going to make changes to your MKB file (particularly in the files or subprojects section) then do this whilst the project is closed down then re-open the project by opening the MKB file again. This will retain your current open files in the IDE.

Memory

Q) What types of memory buffers does Marmalade use?

A) Marmalade uses a variety of memory buffers to retain certain data:

- Heaps – These store generic data that is allocated using malloc and new

- Data cache – The data cache is a scratch area of memory that is reset each frame, you do not need to worry about freeing allocations that you make from the data cache. It is useful for storing temporary data

- Vertex cache memory – Vertex cache memory is used to store temporary transformed vertex data

- Texture memory – Texture memory is usually located in VRAM (in some cases in device memory) which is usually a very limited resource. As such using texture memory has to be planned and tested carefully. Unfortunately Marmalade does not provide any sort of way to check the amount of texture memory available on the device to the application.

Q) What does Heap 0 mean?

A) Heap 0 is the default heap that Marmalade automatically assigned to your application when it boots. Heap 0 is where all memory allocations allocate memory from by default.

Q) How do I increase the size of Heap 0?

A) Edit your projects app.icf file and add or change the MemSize option:

```
[S3E]
MemSize=20485760     # This reserves 20MB
```

Be careful not to be get over zealous when reserving memory as some devices will kick your whole app out of memory and shut it down if it gets low on memory.

Q) Can I use more than one heap?

A) Yes, Marmalade supports up to 8 heaps (heap 0 to 7). You can specify each individual heaps size in your projects app.icf file, e.g.:

```
[S3E]
MemSize1=4194304     # This reserves 4MB for heap 1
MemSize2=4194304     # This reserves 4MB for heap 2
MemSize3=4194304     # This reserves 4MB for heap 3
MemSize4=4194304     # This reserves 4MB for heap 4
MemSize5=4194304     # This reserves 4MB for heap 5
```

You can change which heap is currently being used for allocations using s3eMemorySetInt(S3E_MEMORY_HEAP, heap_number);. Note that you do not need to switch back to a specific heap to free memory allocated from that heap.

Q) Why does my app assert when the Marmalade simulator exits. It show an error similar to "Heap 0 out of memory. Allocating 2048 bytes but only 576 available (540 LFB). Increase icf setting [s3e] MemSize (current value is 14584)"

A) Marmalade's memory manager pre-allocates a specified amount of memory from the system (3MB by default) for your application. When you allocate memory within your app you do not request memory from the operating system itself, instead you request memory from the Marmalade memory manager. This means that you need to approximate the maximum amount of memory that your app can possibly use during its lifetime and set this in the applications configuration file, e.g.:

```
[S3E]
MemSize=20485760     # This reserves 20MB
```

Q) Why does my app assert whilst rendering. It shows an error similar to "IwAssert failure (GX, 761). Message: Data cache overflow allocating 336. Increase [GX] DataCacheSize (currently 800) Callstack: IWGXFNI_HW_DrawPrims_TSW_IwGxInitPipeline"

A) By default Marmalade allocates 16k for the data cache, which is usually much too small for most gaming apps. You can change the amount of memory reserved to the data cache by adding the following to your app.icf:

[GX]
DataCacheSize=102400   # Allocate 100k for the data cache

Q) When my app exits it asserts with an error message similar to "IwAssert failure (MEMORY, 2356). Message: Bucket 0(System) is still in use (first allocation ID=21009); deleting it is dangerous Callstack: IwUtilTerminate IwGxTerminate_IwGxInitPipeline"

A) This assert is caused by an heap imbalance. Basically Marmalade has some items left on the heap that have not been deleted when the app is shut down. Check to ensure that you have deallocated all memory that you have allocated during the apps lifetime.

Q) My app asserts with an error message similar to "IwAssert failure (MEMORY, 1569).Message: Attempting to free item 0x0999AF48, which has either never been allocated, already been freed or was allocated before IwMemBucketInit was called Callstack:_IwGxInitPipeline"

A) This assert means that you are trying to deallocate some memory that has not yet been allocated or more likely already been deallocated. To reduce the chance of this problem occurring you should set pointers to NULL by default, set them to NULL after you deallocate the memory that they point to and always check that a pointer is not NULL before deallocating it. This simple technique will save you so much time and frustration.

Resource Groups

Q) What is a Marmalade resource group?

A) A Marmalade resource group is a collection of assets in a raw binary format. Asset data is stored in a format that is optimised for the target operating system / device. For example, PNG's that are added to the resource group are stored in a raw uncompressed texture format. Resource groups are defined using a resource .group file. A resource .group file is a text file that contains the resources that you wish to be included in the resource collection. During an x86 Debug build run, the Marmalade system will take the resources specified in the .group file and create a binary representation that is stored in the data-ram folder. For example, data\test.group becomes data-ram/build_style/test.group.bin. The default build_style is data-gles1.

Q) Why is my group.bin file much bigger than the data I added to it?

A) Data that is stored within a group binary file is stored uncompressed be default. You can tell the system to compress all data in your group bin files automatically by adding auto-derbh to the deployments section of your MKB file. However, most platforms usually compress their application data when you create a deployment file. For example when a final APK is built for Android the assets within the APK are compressed.

Q) How can I reduce the size of my group.bin files?

A) There are many things you can do to reduce group binary file sizes, including reducing size / colour of textures, reducing polygon / vertex counts of models, reducing animation frames, reducing levels of details, compressing textures etc.. You can also ask the system to compress your assets using the derbh option. Note that sometimes compressing textures using formats such as ATITC / PVRTC can increase your binary file sizes as the compressed textures may not zip better than the original uncompressed image. Also note that storing textures in hardware compressed formats can lead to image degradation and corruption / removal of alpha components as these formats are usually lossy.

Q) Which is fastest, loading an image from a resource group or loading a PNG directly?

A) Generally it is significantly faster to load an image from a resource group as it is uncompressed or can be decompressed quicker than decoding a PNG file.

Graphics / Display

Q) What image formats does Marmalade support?

A) BMP, GIF, PNG, TGA and JPEG

Q) Does Marmalade support software and hardware rendering modes?

A) Yes, you can specify which mode to use by adding one of the following to your app.icf file:

```
[GX]
Rasterisation=HW    # Hardware only
Rasterisation=SW    # Software only
Rasterisation=SW/HW # Supports both but start in software mode
Rasterisation=HW/SW # Supports both but start in hardware mode
```

Q) How do I fix the device's screen orientation to portrait or landscape?

A) Edit the projects app.icf file and add one of the following:

```
[S3E]
DispFixRot="FixedLandscape"                  # Landscape
DispFixRot="FixedPortrait"                   # Portrait
```

Q) What is Iw2D?

A) Iw2D is Marmalades basic 2D rendering API that sits on top of the lower level IwGx API. Iw2D allows you to get 2D graphics up onto the screen quickly and effortlessly. Iw2D takes care of the building of vertex streams, materials creation, building transforms, batch rendering for increased speed etc.. Iw2D supports the rendering of basic primitives such as images, text, arcs, rectangles and lines. Iw2D is the best API to use if you want to get a basic 2D game up and running quickly and you do not require a lot of custom control over rendering.

Q) What is IwGx?

A) IwGx is the low level rendering API used by the Marmalade SDK that sits on top of OpenGL. IwGx allows you the lowest level of control over the graphics system such as building and draw polygons / vertex streams, setting up materials, defining lighting, applying effects, using shaders etc.. IwGx is the best option for seasoned developers who want more control over what and how things get rendered. IwGx usually makes a great base for porting apps to Marmalade as well as creating your own 2D and 3D rendering systems.

Q) What is IwGL?

A) IwGL is a helper API that allows developers that are already accustom to working with OpenGL to port their products to Marmalade without the need to rework all of their existing GL rendering code. IwGL provides features such as EGL initialisation, State caching, Virtual resolution, Calling extensions directly, Consistency across platform and Future proofing. IwGL is ideal for developers that want to quickly port their existing Open GL products to run across multiple platforms using the Marmalade SDK but without having to rework their graphics pipelines.

Q) What is IwGraphics?

A) IwGraphics is a high level 3D rendering API that deals with building and rendering 3D models. IwGraphics can also interface with IwAnim (Marmalades animation API) to provide skeletal based animations out of the box. IwGraphics is built on top of IwGx to ensure optimised rendering. IwGraphics is a good fit for those that do not currently have a 3D rendering engine. Note that Marmalade provides 3D data exporters for Collada, Max and Maya.

Q) Which is the best option for me, Iw2D, IwGx, IwGL or IwGraphics?

A) This all depends on what you want for your app / game. If you are producing an app and not a game then you are best using Iw2D and IwUI or IwNUI. If you are creating a high powered 3D game and you do not have an existing engine then you should go with IwGraphics or IwGx (note that if you go with IwGx you will need to perform all of your own model database management and rendering). If you already have an OpenGL engine then you should probably use IwGL, although I personally would take the time to port everything to IwGx. If you are interested only in 2D games then Iw2D is definitely worth a look. If you want to create 2D games and have full control over rendering then IwGx would be the way to go.

Q) How do I initialise / clean up the various Marmalade rendering API's and which subprojects would I need to include in my MKB file?

A) Note the following function pairs / subprojects

- Iw2D – Iw2DInit(), Iw2DTerminate() - iw2d project
- IxGx – IwGxInit(), IwGxTerminate() - iwgx project
- IwGL – IwGLInit(), IwGLTerminate() - iwgl project
- IwGraphics – IwGraphicsInit(), IwGraphicsTerminate() - iwgraphics project

Q) Why do I see black or coloured lines around the edges of my sprites?

A) This could by caused by a combination of mip-mapping and image filtering. Iw2D in particular will generate mip-maps for all images that you create and use. When filtering is enabled the chosen pixel will be an average of the surrounding pixels from the chosen mip-map. To reduce the chance of seeing this problem disable mip-mapping for Iw2D using Iw2DSetUseMipMapping(false);

Q) I STILL see black or coloured lines around the edges of my sprites, what gives?

A) Try removing the opacity / alpha channel from your image and filling the transparent areas with pure magenta (Red = 255, Green = 0, Blue = 255). Marmalade uses pure magenta as a colour key representing fully transparent areas. GIMP has a nice tool for quickly removing the opacity / alpha channel from an image called "Flatten Image", located in the Image menu. Once the opacity / alpha channel has gone you can flood fill the cut out areas of the image with pure magenta

Q) Why do my textures that contain pure magenta show up the magenta areas as transparent?

A) Marmalade use pure magenta (Red = 255, Green = 0, Blue = 255) as a colour-key to allow images with no transparency to have fully transparent area. This is great for sprites as you can store a full colour image without the transparency channel, reducing the amount of texture RAM required to render it and increasing rendering speed, however this effect cannot be turned off. An easy work

around is to use a magenta that is a slightly different shade of magenta.

Q) I am trying to mix raw Open GL rendering code in with iwGx / Iw2D but I cannot see anything rendered from Open GL, how do I fix it?

A) Replace Marmalades screen clearing function IwGxClear() / Iw2DSurfaceClear() with the following:

```
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
```

Q) I am mixing raw Open GL rendering code with IwGx but all of my polygins appear below all of the IwGx rendered polygons, why does this happen and how do I swap the order?

A) IwGx render calls are not executed until you call IwGxFlush(). During the frame you make calls to IwGx to render polygons then make calls to GL to render polygons with GL. The calls you make to GL are rendered immediately whereas the calls to IwGx are rendered at the end of the frame when IwGxFlush() is called, hence IwGx polygons are drawn on top of your GL polygons. To solve draw all of your IwGx content, call IwGxFlush() then draw all of your Open GL content.

Q) I get odd things happening after making calls to IwGx / Iw2D then call raw Open GL code. Whats happening and how do I fix it?

A) IwGx and Iw2D are based on Open GL so they change Open GL's internal states. The best solution is to set up your own reliable default states after you have finished rendering IwGx / Iw2D content. Particular areas that are affected include matrices including texture matrix and shading model.

Fonts

Q) How do I create fonts with the Marmalade SDK?

A) There are currently two ways to create fonts:

a) The Marmalade SDK ships with a tool called Marmalade Font Builder that allows you to generate bitmap fonts that can be rendered. This tool will export a low colour TGA and a .gxfont file. The associated TGA file contains the graphical data for each glyph whilst the .gxfont file contains information relating to which characters are supported as well as other info.

b) You can add true type fonts directly to a Marmalade resource group or load them directly using

IwGxFontCreateTTFont() / wGxFontCreateTTFontFromBuffer(). Below is an example showing how to generate two different sized versions of a font from a TTF

```
CIwGxFont
{
        name "Serif_6"

        ttf ".\Fonts\serif.ttf"
        ttfPointSize 6
}

CIwGxFont
{
        name "Serif_8"

        ttf ".\Fonts\serif.ttf"
        ttfPointSize 8
}
```

You can then load these fonts in your app using:

```
// Load the group containing the example font
IwGetResManager()->LoadGroup("IwGxFontTTF.group");

// Get the example font and set it to be the current font
CIwGxFont* s_Font6 = (CIwGxFont*)IwGetResManager()->GetResNamed("Serif_6", "CIwGxFont");
CIwGxFont* s_Font8 = (CIwGxFont*)IwGetResManager()->GetResNamed("Serif_8", "CIwGxFont");
```

Q) I've added a true type font to my resource group but I get an assert that says "TTF support requires IW_GXFONT_TTF to be defined in the project's mkb before including the iwgxfont subproject." when I attempt to tun my app. What do I do?

A) To use TTF's in your build you need to add the following to your project MKB file:

```
defines
{
    # Define for TTF rendering support.
    # Must be before iwgxfont subproject is first included.
    IW_GXFONT_TTF
}
```

Q) Do I need to license a font to use it in my products?

A) Yes, like any other asset in your app, fonts cost money to make and maintain, hence many are not free. However there are plenty of fonts out there that are free or have relaxed licensing restrictions. Take a look at http://openfontlibrary.org/ as there are many fonts here that allow embeddding in your products (always check the license before use though).

Q) Which is fastest, rendering bitmapped fonts or rendering true-type fonts?

A) Initially rendering bitmapped fonts is much faster as the glyphs are already stored in the format in which they will be rendered. True type fonts on the other hand are built in real time and glyphs are cached as and when required.

Q) I've added a true type font to my app but I get the following assertion similar to "Out of cache for font 'Serif_8'. Use the 'ttfCacheSize' font conditioning setting to increase cache size." when I run my app, how do I fix this?

A) Marmalade reserved by default enough space to cache 192 different characters for each font that is loaded. If your font uses more than 192 glyphs then you need to increase the number to allow more glyphs to be cached. To increase this you need to add the following setting to your app.icf:

```
[GxFont]
CacheTextureMaxSize=512
```

This will increase the maximum size of the texture used to store glyphs making room available for more.

Audio

Input

File System

Communications

Video

Q) Does Marmalade SDK support video playback? If so what types of formats does it support?

A) Yes, Marmalade supports MPEG4 and 3GGP. As for which device supports which format that varies from device to device. You can use either s3eVideoPlay() or s3eVideoPlayFromBuffer() to play the video file

Q) How do I determine if a particular device supports a particular codec?

A) You can call s3eVideoIsCodecSupported(s3eVideoCodec codec); to determine if the device supports a particular codec

Q) Can I play video full screen?

A) Yes, when you call s3eVideoPlay() or s3eVideoPlayFromBuffer() you can pass the top, left coordinate of where the video should be placed as well as the width and height.

Q) Can I display overlays over the top of the video?

A) No, not at this time

Q) Can I change the play back position of a video file, for example, start from 2 minutes in or go back 1 minute in?

A) No, at the moment you can only play, pause and resume video play back

Marmalade Simulator

Q) What is the Marmalade simulator?

A) The Marmalade simulator is an application that simulates how your application will look, run and feel to the user on an actual device. The simulator simulates many things including different device orientations and screen sizes, input devices, audio, device events and more. The simulator also provides useful metrics which you can use to track things such as how much memory is used, how memory is fragmenting, how much texture RAM you have used etc..

Q) Occasionally when I run the Marmalade simulator it runs at a very slow frame rate (as low as 1 fps), how do I solve this?

A) This can occur from time to time when your computer is low on memory, you may also notice some texture corruption, especially when rendering fonts. Close down some apps and try running the simulator again.

Q) There is significant reduction in quality in my images such as colour banding when I display them on the simulator, how can I fix this?

A) The problem is that the simulator is using a lower colour rendering surface such as RGB565. You can increase the surface colour depth by going to Configuration->Surface and changing Device Pixel Format to RGB888

Q) I am seeing bad artefacts in text and images in the simulator, what could be causing this?

A) Go to Preferences->Display and check that Display Width / Height Scale are both set to 1. If set to a value smaller than 1.0 then text and images tend to be display compressed without filtering.

Q) How do I use Metrics in the simulator?

A) Firstly enable Metrics by selecting Metrics->Enable Metrics. Now go to Metrics->Display Metrics. Metrics lets you view all sorts of useful information about your app, such as how many memory allocations have happened each frame, how much texture space is used, how many polygons have been rendered / clipped etc..

Deployment

Q) I deployed my game / app to my device but it crashes. I checked the debug output and noticed that my game assets that I placed in a resource group were not found. How come?

A) Before you deploy your build to an actual device you need to run the x86 Debug build in the Marmalade simulator. The x86 debug build takes your raw assets that are located in resource group files and converts them to an optimised binary format that is better suited for the target operating system / device. These files are usually generated in the data-ram folder. Ensure that you add these generated binary files into the assets section of your MKB, e.g.:

```
assets
{
        (data)

        (data-ram/data-gles1, data)
        fonts.group.bin
}
```

Q) I deployed my game / app to my device with changed assets but my assets didn't change, what could be the cause?

A) There are a number of reasons for this. The first is that you forgot to run the x86 debug build to regenerate your resource groups or the app was not installed because it was already present on the device. Some devices such as iOS devices do not update the app automatically unless the version number has increased. If you do not mind losing any data generated by the app then you should remove the previous version of the app before installing the new version

Q) Can I deploy to iPhone, iPad or iPad without a Mac?

A) Yes you can, however you will need a Mac to perform the final app submission to Apple for approval

Q) I'm having trouble deploying my build to an iOS device and / or Apple iTunes App store, where can I get help?

A) DrMop provides a complete step by step walk-through covering iOS provisioning, iOS certificates, device and app store deployment at
http://www.drmop.com/index.php/2011/12/10/marmalade-sdk-tutorial-apple-iphone-and-ipad-deployment-and-submissions/

Q) I'm having trouble deploying my build to a BlackBery PlayBook device and / or BlackBerry App World store, where can I get help?

A) DrMop provides a complete step by step walk-through covering preparing your system, generating / installing debug tokens, deployment to device and the BlackBerry AppWorld store at
http://www.drmop.com/index.php/2011/08/30/marmalade-sdk-and-blackberry-playbook-from-setup-and-deployment-to-app-world-submission/

Porting

Q) Can I port my Open GL based app / game to Marmalade?

A) Yes, you can use OpenGL functions directly as is within Marmalade as well as use the IwGL helper API to integrate your code better into Marmalade. That said IwGx, Marmalades low level rendering API is close to OpenGL so its not much additional work to convert your code to operate directly under IwGx's control. By opting to convert your OpenGL code to IwGx you get additional benefits such as automatic batching, asset optimisation, better compatibility with more devices and of course, better future proofing.

Q) Is there anything I should do to my code base before converting it to use the Marmalade SDK?

A) The most optimal approach is to ensure that all functionality relating to platform specifics such as rendering, input systems, audio, file system access, sockets and http access etc.. are abstracted into their own hardware abstraction layer, so basically no mixing platform specific calls in with your general code. This way you only have a small number of files to convert to Marmalade. You could even include stub classes / functions for unimplemented features so you can test the port as you go along.

Middleware

Q) Are there any free open source game engines available that are compatible with Marmalade?

A) Yes, you can check out the IwGame Engine and Cocos2D. Both engines are open source and free to use.

Q) Are there any free open source application engines available that are compatible with Marmalade?

A) Yes, you can check out the IwGame Engine which supports a rich extensible mark-up driven UI and layout system

Q) Is there an official list of open source projects that I can use with the Marmalade SDK?

A) Yes there are a large collection of open source projects available. Including projects such as Box2D physics, AdMob, Tapjoy, Flurry, Facebook, AdWhirl, TinyXML, Bullet physics, sqlite and many others. A full list is available at https://github.com/marmalade