

Vulkan & OpenGL

AppGameKit Studio features both OpenGL and Vulkan rendering engines.

The OpenGL engine works on all supported platforms, it uses OpenGL 2.0 on Windows, Mac, and Linux, and OpenGL ES 2.0 on Android, iOS, and HTML5.

The new Vulkan engine works and is now the default on Windows, Mac, Linux, and iOS. On iOS and Mac the Vulkan renderer runs on top of Metal using the MoltenVK wrapper, whilst it is not a native implementation of Vulkan it should perform the same as Vulkan on the other platforms.

Android has partial support for Vulkan, currently `PlayVideoToImage`, `SetDeviceCameraToImage`, and the AR commands are not supported with Vulkan. Therefore the Android player will always run OpenGL ES 2.0 to allow maximum compatibility with apps broadcast to it. If you do not use any of these commands then you can choose to use the Vulkan renderer in your exported apps using the **#renderer** "Advanced" line in your Tier 1 code as detailed below.

Choosing The Renderer

To force AGK to use a particular renderer you can use **#renderer** followed by a string to tell AGK which renderer you want to use. The valid strings are

- "Basic" to use OpenGL 2.0 or OpenGL ES 2.0 depending on the platform, if this is not supported the app will close
- "Advanced" to use Vulkan, if this is not supported then the app will close (except on Android where it will then fallback to OpenGL ES 2.0)
- "Prefer Best" will try to use Vulkan and fall back to OpenGL if Vulkan is not supported by the current device

This can be used at any point in your Tier 1 code, it will be read by the compiler and applied when your app first starts.

The renderer cannot change once it has been set up, so **#renderer** will do nothing when broadcast to a player, since the player will already have started and chosen a renderer before it runs your Tier 1 code. The **#renderer** option only applies to apps run using the Run button in the IDE, apps exported from the IDE, or apps run directly from their project folders.

Shaders are automatically converted to a format suitable for the current renderer. All shaders must be in the GLSL V110 format, if the renderer is OpenGL or OpenGL ES then they are used as is in GLSL format. If the renderer is Vulkan then AppGameKit Studio converts them into SPIRV format or Metal Shading Language as appropriate.

If you experience any problems running your apps on OpenGL or Vulkan hardware then please report your issues to our team here: [AppGameKit Studio Issues](#)