

Introduction to CMake (8A)

Will Pirkle

This document explains why RackAFX is now using CMake to package exported projects as well as information about how to generate both individual and universal exported plugin project solutions. Even if you already know how to use CMake, you should still familiarize yourself with this document to understand any particulars involving the RackAFX exported projects.

Contents:

1. What is CMake?

1.1 Why the change to use CMake? Why are you making us use it?

2. Individual Projects vs. Individual Solutions

2.1 Individual API Projects

2.2 Universal API Projects

1 What is CMake?

CMake is a utility that allows you to create a Visual Studio or Xcode project using a simple text based script. You can also create projects for many other compilers for Mac, Windows and Linux. CMake is free and you can get a copy here:

www.cmake.org

CMake is most efficiently used with the Command Prompt (Windows) or Terminal (MacOS), although it does have an optional GUI interface. Fortunately, there are few commands to type and since the GUI may change at any time, only the command prompt version is documented here (trust me, it is really easy).

1.1 Why the change to use CMake? Why are you making us use it?

CMake solves many problems with distributing code that is designed to be used in a project that requires an external set of SDK sources (files) and other resources. The architects of the various SDKs typically give themselves some freedom in making underlying changes in the SDK itself, as long as there is no impact (or at least very little impact) to your non-SDK source files. The architects will make these changes - adding, deleting or moving files, adding or obsoleting objects and functions, etc... on a semi-regular basis. This is all in the name of progress. But in general, for some manufacturers you are virtually guaranteed that a VS or Xcode project compiled with the previous SDK, will fail to compile with the new version. Usually it is just due to the relocation of a file or some kind of changes to the various compiler settings, but it can be very frustrating.

Currently, Steinberg packages the VST3 SDK using CMake. When they make changes to the SDK, they also change the CMake script files. Then, it is your job to run CMake on the new SDK which sets up the sample code - every sample project is packaged inside of one gigantic VS or Xcode project. By copying the CMake files, you can easily create spin-off projects to practice with. When you use the new RackAFX exported packages and CMake, you can easily move

your project to the new SDK, re-run CMake, and be updated automatically with the SDK changes. This removes a lot of frustration when upgrading SDKs.

Currently, neither the AAX nor the AU SDK is packaged with CMake, however the RackAFX exported versions are designed so that this has minimal impact on them.

★ **Projects Designed for Your Compiler/Platform**

A major benefit of using CMake to build your own project files, rather than relying on pre-made files, is that CMake will target the exact compiler you have installed on your system and create a project specifically for it. What's more is that you can write the scripts for both the Windows VS and Mac Xcode projects in the same files when needed, and in separate files when that is warranted.

★ **Easily Create Universal API Configurations**

The second benefit is that your project files may be located wherever you wish relative to the SDKs themselves. In previous versions of RackAFX, the AAX and VST projects needed to be located at a specific folder hierarchy level with each SDK. With v6.9, your exported projects are packaged for a default location within the SDKs - OR - **outside the SDKs altogether** in a Universal API Configuration. If you want to change the project location, you only need to change a couple of lines of text inside a single CMake file (see the accompanying *Using CMake* document).

2 Individual Projects vs Universal Solutions

2.1 Individual API Projects

The most basic way to use your RackAFX exported project is within a single API. For example, you might prefer to only write VST plugins because you don't care about AU or AAX. When you use the *Export Project* function in RackAFX and choose an "Individual" VST project, it will create a folder with only the VST project enabled for CMake. After running CMake, you will have a single VST project to work with in either Xcode or Visual Studio.

2.2 Universal API Projects

If you want to create only a single Xcode project that contains the MacOS VST, AU and AAX version of your plugin, or a Visual Studio project that contains the Windows VST and AAX versions, you can now easily do this with RackAFX. When you use the *Export Project* function and choose "Universal" in RackAFX, it will create the same folder as for the individual projects, but with **all APIs that you choose** enabled for CMake. After running CMake, your Xcode and VS solutions will contain individual projects for each API. With two *ALL_BUILD* commands, you will have every version of your plugin for both Mac and Windows.

2.3 VSTGUI4

The RackAFX exported projects use VSTGUI4 as the GUI design and packaging mechanism. The new v6.9 exported projects also require it. The project folder you initially setup must contain the same VSTGUI4 sub-folder version that RackAFX uses for the current exports.

Important: as with the VST SDK, the VSTGUI4 architects also make fundamental changes to the SDK from time to time which may affect your project code. New VSTGUI4 releases are always timed to coincide with new VST3 SDK versions. Because the SDK is changed often, the RackAFX exported projects use their own version that is proper for the current release. This

includes the VST exported projects. In this way, if the VSTGUI4 SDK has a substantial change, you can still compile with the latest VST3 SDK without the changes affecting your plugin. In addition, it keeps the RackAFX exports and their supporting files out of the way of the SDK itself and easily movable when new SDKs are released, without requiring any previous SDK files or folders.

You can get the latest (or proper) VSTGUI SDK files from various locations (see below) including the Forum at www.willpirkle.com and even from RackAFX itself. The Utilities -> Install VSTGUI4 option will install a copy of the complete SDK in the folder of your choice. It will always copy the currently compatible version. You can find the SDK at GitHub here:

<https://github.com/steinbergmedia/vstgui>