

Exporting RackAFX Projects for CMake (8B)

Will Pirkle

This document explains how to export AAX, AU and VST projects with RackAFX v6.9 and above. It includes the details of setting up your target API/SDK, folder naming conventions, and the locations of your finished exported projects. It details the differences between exporting **individual** projects and **universal** projects.

Contents:

- 3. Folder Naming Convention
- 4. Exported Project Location Overview
- 5. Individual Exported Project Locations
 - 5.1 AU
 - 5.2 AAX
 - 5.3 VST
- 6. Universal Exported Project Location

3. Folder Naming Convention

You will need to navigate either Terminal or the Windows Command Prompt to various locations within your SDKs to run CMake. To keep the folder names simple and easy to use with Terminal/Command Prompt, I use a naming convention with:

- no whitespaces (blanks)
- no numbers
- no special characters

You will see this echoed throughout the document and *readme.txt* file that arrives with each exported project.

4. Exported Project Location Overview

For both the individual and universal API projects, there is only a single outer folder that contains all the RackAFX exported projects along with a single copy of the VSTGUI4 sub-folder. I use the folder name “*myprojects*” for all of my exports, so I will also use it here as an example. **Within your outer *myprojects* folder, you need to include the VSTGUI4 sub-folder.** A typical *myprojects* folder that contains the VSTGUI4 sub-folder along with three exported projects named *Compressor*, *Reverb*, and *ParametericEQ* would have the following folder hierarchy:

- ▼ myprojects
 - ▶ vstgui4
 - ▶ Compressor
 - ▶ Reverb
 - ▶ ParametericEQ

Expanding the *ParametericEQ* folder reveals the first layer of project files

- ▼ myprojects
 - ▶ vstgui4
 - ▶ Compressor
 - ▶ Reverb
 - ▼ ParametericEQ
 - ▶ cmake
 - ▶ mac_build
 - ▶ project_source
 - ▶ win_build
 - CMakeLists.txt

Inside of this folder is the **outermost** CMakeLists.txt script file. If you ever need to change the location of your projects relative to the SDK, this is the only file you need to edit (see the accompanying *Using CMake* document). You can take some time and poke around in the various subfolders, but you will generally only need to work inside of the *mac_build* or *win_build* folders when creating MacOS or Windows plugins respectively. You can find your original RackAFX project code inside of the folder:

```
../project_source -> source -> rafx_source
```

5. Individual Exported Project Locations

There are three supported APIs for RackAFX: AAX, AU and VST and each SDK consists of a series of subfolders with AU being the simplest and VST being the most complex in terms of folder hierarchies.

5.1 AU

The AU SDK is contained within two subfolders named *AUPublic* and *PublicUtility*. You can get these two folders along with the latest sample code here:

<https://developer.apple.com/library/content/samplecode/sc2195/Introduction/Intro.html>

or

www.willpirkle.com/Downloads/AU_SDK.zip

For the AU SDK, you first need to create an outer container for the two SDK folders. You may name this folder whatever you wish - I use the name "*AU_SDK*" for my folder so we will use it here. For the RackAFX ported projects.

Place your *myprojects* folder in parallel with the two SDK files and inside of your outer *AU_SDK* folder like this:

- ▼ AU_SDK
 - ▶ AUPublic
 - ▶ **myprojects**
 - ▶ PublicUtility

At this point, you are ready to start exporting AU projects from RackAFX. And, if you decide to use the Universal API paradigm instead, you can re-use the same *AU_SDK* outer folder without issues.

5.2 AAX

The AAX SDK is contained within a series of subfolders of the current SDK branch. You must establish an account with Avid to gain access to the SDK and Pro Tools Developer Version and you will need an iLok2 device for storing your Pro Tools activation key.

The AAX SDK architects use numbers when naming the various versions such as `AAX_SDK_2p3p0` which reads “AAX SDK version 2.3.0.” I prefer to rename the current SDK to `AAX_SDK` in keeping with folder naming convention discussed in the *Intro to CMake* document, as well as shortening the name a bit. As with the AU projects, your *myprojects* subfolder will be located in parallel with the other SDK sub-folders at the outermost level. My folder hierarchy looks like this:

- ▼ AAX_SDK
 - ▶ Documentation
 - ▶ ExamplePugins
 - ▶ Extensions
 - ▶ Interfaces
 - ▶ Libs
 - ▶ **myprojects**
 - ▶ TI
 - ▶ Utilities

At this point, you are ready to start exporting AAX projects from RackAFX. And, if you decide to use the Universal API paradigm instead, you can re-use the same `AAX_SDK` outer folder without issues.

5.3 VST

The VST SDK is contained within a series of subfolders of the current SDK branch. The architects name the outermost folder `VST_SDK` and it contains two inner folders, `VST2_SDK` and `VST3_SDK`. In addition, there are two script files, one for Mac and one for Windows. These scripts install the VST2 SDK folders within the VST3 SDK so that you may use the VST2 wrapper that is included with the VST3 SDK.

Important: all RackAFX exported VST projects are VST2 compatible and require the VST2 API files to be setup properly. This is fairly simple:

Windows: Navigate to the `VST3_SDK` folder that contains `copy_vst2_to_vst3_sdk.bat`, then, double-click on the `copy_vst2_to_vst3_sdk.bat` file to run the copy mechanism.

MacOS: Open a Terminal window and navigate to the `VST3_SDK` folder that contains the `copy_vst2_to_vst3_sdk.sh` file. Drag and drop the `copy_vst2_to_vst3_sdk.sh` file into Terminal. This will run the copy mechanism.

With the VST2 SDK installed, you can now create the *myprojects* folder for your exported projects. It needs to be located just inside the VST3_SDK subfolder, in parallel with the other SDK files, exactly as with AU and AAX. Your folder hierarchy should look like this:

- ▶ VST_SDK
 - ▼ VST3_SDK
 - ▶ base
 - ▶ bin
 - ▶ cmake
 - ▶ doc
 - ▶ **myprojects**
 - ▶ pluginterfaces
 - ▶ public.sdk
 - ▶ vstgui4
 - CMakeLists.txt
 - index.html

You can see the SDK's CMakeLists.txt file - this is entirely independent of your exported projects and when you run CMake for the exported project, it will not interfere in any way with the SDK's files or CMake scripts. You should note that *index.html* is the Doxygen documentation link.

At this point, you are ready to start exporting VST projects from RackAFX. And, if you decide to use the Universal API paradigm instead, you can re-use the same *VST_SDK* outer folder without issues.

6. Universal Exported Project Location

Creating a folder hierarchy for the Universal setup is simple: create an outer folder to hold the SDK sub-folders in parallel with each other. I use the folder name *ALL_SDK* for my Universal projects. Your *myprojects* sub-folder is located in parallel with all three SDK folders and is **not** inside any one of them. My universal project folders look like this:

- ▼ ALL_SDK
 - ▶ AAX_SDK
 - ▶ AU_SDK
 - ▶ **myprojects**
 - ▶ VST_SDK

At this point, you are ready to start exporting UNIFIED projects from RackAFX. You can also use the individual projects contained with each SDK without any interference from the other

projects. It is also simple to edit the outermost CMakeLists.txt file to convert individual exported projects to universal ones and back again by changing a few lines of script - see the accompanying *Using CMake* document.