Upgrading Projects and VSTGUI4 Library
for
RackAFX 7.0.3.1 with ASPiK 1.6.8 Export

These instructions apply to RackAFX 7 users, for updating to the new RackAFX 7.0.3.1 which exports ASPiK 1.6.8 plugin projects.

Prior to updating:
Make a backup copy of your RackAFX7 project folder including the vstgui4 library folder. This is a safety measure in case you need to go back to a previous build.

VSTGUI4 -> 4.10
A significant change was made to VSTGUI4 for the new version 4.10 that involves the initialization of the library. This is completely transparent but required me to change some fundamental code for the GUI and for exporting.

When you run RackAFX 7.0.3.1 for the first time, it will detect your older version and ask to install v4.10. You need to do this to create new projects with RackAFX 7.0.3.1. You may always install it later from the Utilities -> Install VSTGUI4.10 menu item.

RackAFX NEW Projects
I added new template code to make it easier for you to process audio in complete buffers, or in sub-blocks operating directly on the DAW buffers for maximum efficiency and plugin speed. This capability has always been in the ASPiK core, but required users to write that code and many requested that I add the template code.

When you create a new project, it will default to <u>frame processing</u> as normal. If you want to process buffers or sub-blocks, you can choose that when you setup the project, or later in code, or you can use File -> Edit Project to change the settings.
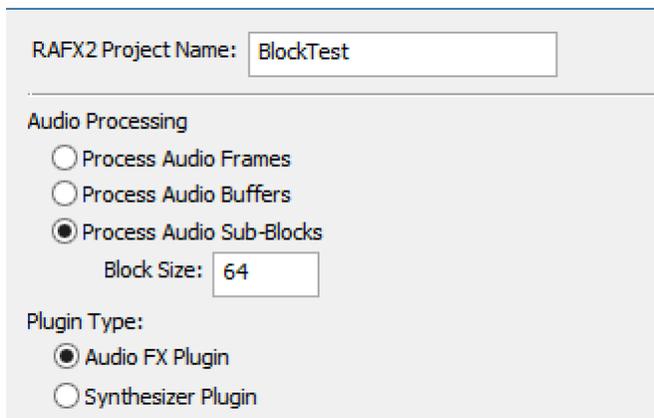
The new project dialog includes selections for frames (default), buffers or sub-blocks, which are used in my new Synth book projects. You may also set the sub-block size in this area of the dialog box.

Changing the Frame/Buffer/Block Processing in Code

Your plugindescription.h file will include two constants that define the audio processing operations named kProcessFrames and kBlockSize. You can edit these lines of code directly to alter the processing mechanism.

Process Frames:

To process frames (default) set **kProcessFrames = true**. The block size variable will be ignored:

```
// --- Plugin Options
const bool kProcessFrames = true;
const uint32_t kBlockSize = DEFAULT_AUDIO_BLOCK_SIZE;
```

Process Entire Buffers:

To process whole buffers, set **kProcessFrames = false** and set the block size to the constant **WANT_WHOLE_BUFFER** or its actual value of zero (0).

```
// --- Plugin Options
const bool kProcessFrames = false;
const uint32_t kBlockSize = WANT_WHOLE_BUFFER;
```

Process Sub-Blocks

To process sub-blocks, set **kProcessFrames = false** and set the block size to the value you want; this is the number of samples per channel (or number of frames) that you wish to process; the default size is 64 which has a specially defined constant DEFAULT_AUDIO_BLOCK_SIZE or you may type 64 directly. You normally want the sub-block to be smaller than the DAW buffer size, and a power of two like 32, 64, 128, 256, etc...

```
// --- Plugin Options
const bool kProcessFrames = false;
const uint32_t kBlockSize = 64;
```

RackAFX EXISTING Projects

When you open an older RackAFX project, it will automatically copy over the new files you need (after it makes backups of your old ones). You must have VSTGUI4.10 installed, then perform a full clean and rebuild of the project. It will NOT attempt to

change the code for frame, buffer or block processing as this would be difficult to do programmatically.

NOTE: this is much simpler than migrating the projects manually in ASPiK, which requires that you perform the same copying and file editing that RackAFX will perform on your project.

RackAFX Bundle Packaging
For new projects, RackAFX now packages the plugin in a bundle, the same way as AAX, AU and VST3. The bundle is actually a folder disguised with an icon that makes it appear to be a file (same is true on MacOS).

When you migrate and rebuild **older** projects, they will NOT be packaged this way, and will remain in the older .rafx2 DLL file format. RackAFX 7.0.3.1 and above will detect the kind of packaging and adjust automatically, so it will read either kind of file/bundle.

Your bundle will have an icon, and you can enter into the bundle by double clicking on it - you will find the icon and .ini files, and two folders. Here, I am inside of the "AloneLFO" plugin bundle:
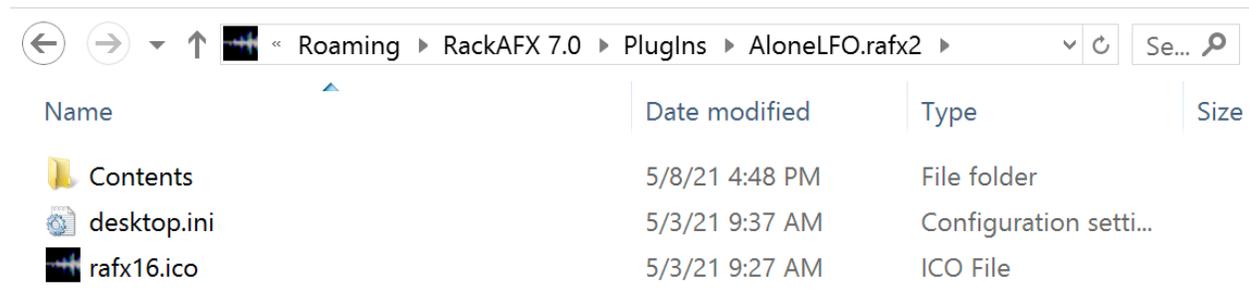
Name	Date modified	Type	Size
📁 Contents	5/8/21 4:48 PM	File folder	
⚙ desktop.ini	5/3/21 9:37 AM	Configuration setti...	
🗠 rafx16.ico	5/3/21 9:27 AM	ICO File	

If you open the Contents folder, you can find the resource and plugin outer containers:
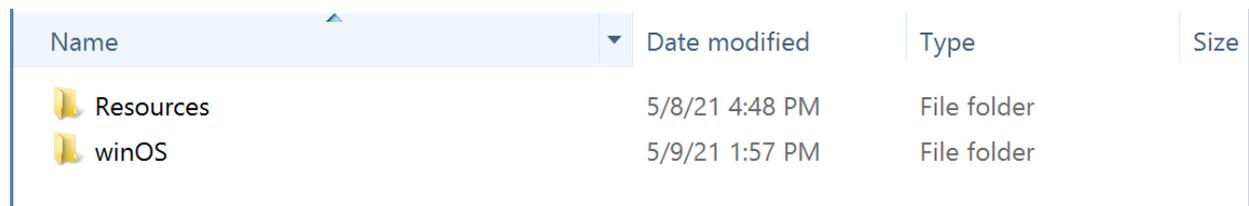
Name	Date modified	Type	Size
📁 Resources	5/8/21 4:48 PM	File folder	
📁 winOS	5/9/21 1:57 PM	File folder	

Inside of the winOS folder, you can find the .rafx2 plugin; here it includes the .pdb file as this was from a debug build:
Packaging in the bundle format will make customization simpler, along with the ability for your plugin to carry around its own payload of files, graphics, scripts, or other things you (or I) might want to add, including 3rd party libraries or DLLs.

Name	Date modified	Type	Size
AloneLFO.pdb	5/13/21 2:28 PM	PDB File	3
AloneLFO.rafx2	5/13/21 2:28 PM	RAFX2 File	4