

## Migrating Projects to ASPiK 1.6.8

These instructions apply to ASPiK users, for migrating older projects to the new SDK. This is the most foolproof method, and preserves your older project while creating a new version in parallel with it. This method will require copying three files, and editing one of them in a couple of single-line locations. It is not difficult to do, and since you will retain your original plugin in an unscathed manner, you can perform the process repeatedly if you make a mistake along the way.

### **Prior to updating:**

Make a backup copy of all of your *myprojects* (or whatever you name them) folders including the vstgui4 library folders. This is a safety measure in case you need to go back to a previous build and one of those “because we are programmers” kind of chores.

### **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.

**Remove/rename your existing vstgui4 folders and copy the vstgui4 folder from the ASPiK SDK. At this point, compiling older projects will fail, as you need to copy some files, make a minor edit to your PluginCore.cpp file, re-run CMake, and then rebuild the project.**

In this example, suppose you have an existing project called **StereoDelay** and you want to migrate it. First, create a new ASPiK project with the ASPiKreator, here I will name it **StereoDelayEx** and will refer to it as the “**new project.**”

### **1) Create the new Project**

Use the ASPiKreator to create the new project in the correct destination for your chosen API(s). Do not run CMake yet.

### **2) Copy existing files into the new project**

Copy the following files from your existing project into the same-named folder in the new project:

*project\_source/resources/PluginGUI.uidesc*  
*project\_source/source/PluginKernel/plugincore.h*  
*project\_source/source/PluginKernel/plugincore.cpp*

### **3) Modify the plugincore.cpp file: Part 1**

In the new project’s PluginKernel folder, open your existing plugincore.cpp file (the one you just copied) in a text editor and scroll to the bottom to alter one single line of code.

Notice that the static return variable has changed to a function named `getPluginDescBundleName()`:

**Old Code:**

```
const char* PluginCore::getPluginBundleName() { return kAUBundleName; }
```

**New Code:**

```
const char* PluginCore::getPluginBundleName() { return getPluginDescBundleName(); }
```

**4) Modify the plugincore.cpp file: Part 2**

You need to change one more function call, named

**`doSampleAccurateParameterUpdates()`** as it has been broken into two functions for faster rendering. Change the call in `processAudioFrames()` as:

**Old Code:**

```
doSampleAccurateParameterUpdates();
```

**New Code:**

```
const char* PluginCore::getPluginBundleName() { return getPluginDescBundleName(); }
```

**Save and close the plugincore.cpp file.**

**5) Run CMake & Rebuild**

Run CMake on the new project using the same method as you did for the original project. Open the compiler project and rebuild the code. this completes the project migration.