

# ISADORA SDK v1.2 INSTRUCTIONS

The distribution of the Isadora SDK for XCode includes three folders inside the isadora-sdk-xcode folder:

- **IsadoraLib** - The static library required to compile an Isadora Plugin
- **IsadoraDemoPlugin** - A demo video plugin
- **IsadoraDemoPluginMIDI** - A demo MIDI plugin

It is important that IsadoraLib folder remains at the same level as your other plugin folders, i.e., you should create your new projects inside the isadora-sdk-xcode folder.

\*\*\* IMPORTANT \*\*\* Because of a bug in XCode, it is required that the folders enclosing the isadora-sdk-xcode folder have no spaces in their names, otherwise the plugins will fail to build. For example the path:

`/Documents/MyCode/isadora-sdk-xcode/`

is OK, while the path

`/Documents/My Code/isadora-sdk-xcode/`

is not. (Note the space between “My” and “Code”)

To create your Isadora plugin, you will start by modifying the Isadora Demo Plugin supplied with the SDK. Before beginning, duplicate the folder called “IsadoraDemoPlugin” (again, keeping it inside the isadora-sdk-xcode folder) and rename the folder it with the name of your plugin. (No spaces in the folder name!)

After making the copy, double click the “Isadora Demo Plugin.xcodeproj” XCode project found in the folder. Then follow the steps below to customize the project.

## **STEP 1: CUSTOMIZING THE PLUGIN NAME**

The first step in customizing this project is to rename the build target. In XCode, do the following:

1. Double click “Isadora Demo Plugin”, which can be seen in the “Groups and Files” section of the project window.
2. Choose **Project > Edit Active Target**
3. In the dialog that appears, click on the "General" tab at the top of the window.
4. Type the name of your plugin in the “Name” field.
5. Now click the "Build" tab.
6. To the right of “Configuration”, choose “All Configurations”.
7. To the right of “Collection”, choose “Packaging”.
8. Under the column marked “Setting”, find “Product Name”. Double-click where you see “Isadora Demo Plugin” and a text edit box will appear.

9. Enter the plugin name in the text edit box and hit return. Don't use any extensions for this name, as PB will automatically add ".izzyplug" the name when the plugin is built.
10. Now click the "Properties" tab.
11. Enter the name of your plugin in the "Executable" field.
12. Close this window.

## **STEP 2: BUILDING THE PLUGIN INTO ISADORA'S PLUGIN FOLDER**

To ease debugging with XCode, you will want to set up so that it builds the plugin directly into Isadora's "Isadora Plugins" folder, which is inside the application package. Follow these steps:

1. In the popup at the top-left of the project window, choose the active configuration, either Debug or Release.
2. Under the "Groups & Files" pane at the left of the window, locate the "Targets" item. Below this it will say show an item with the name of your plugin (e.g., "MyPlugin.izzyplug"). Click the disclosure triangle next to Targets if you can't see this item.
3. Reveal the "build steps" within "Isadora Demo Plugin" by clicking its disclosure triangle.
4. Control-click the plugin name, just below the "Targets" heading. In the popup menu choose **Add > New Build Phase > New Copy Files Build Phase**. A dialog will appear that allows you to specify the path where the built plugin will be placed.
5. To the right of "Destination" choose **Absolute Path**. Then enter the absolute path to your copy of Isadora, plus /Contents/MacOS/Isadora Plugins. For example:  

```
/Applications/Isadora.app/Contents/MacOS/Isadora Plugins/
```
6. Close this window.
7. Finally, from the "Products" header, drag your plugin (e.g., "MyPlugin.izzyplug") into the "Copy Files" build step you just created. This tells the build step to copy the built plugin to the path you specified earlier.

Now, whenever you build the project, the plugin will automatically be copied into Isadora's Plugin Folder.

## **STEP 3: MAKING ISADORA THE APPLICATION TRUN WHEN DEBUGGING**

You also need to set up Project Builder so that it will launch Isadora when you debug.

1. Choose **Project > New Custom Executable**
2. To the right of "Executable" type "Isadora" - this will be the name of the custom executable.
3. In the executable path, enter the path to your Isadora application. For example, if Isadora is in the Applications folder, you would enter  
`/Applications/Isadora.app`
4. Click the "Finish" button. A dialog will appear that allows you to change the settings for the new custom executable.
5. Close this window

## **STEP 4: CUSTOMIZING THIS SOURCE CODE**

In the source file `IsadoraPlugin.cp`, search for each occurrence of the string `###`. All of the places where you will need to customize the file are marked with this pattern of characters.

The source code is heavily commented to help you understand how to modify it for your needs.

The current example plugin projects included in the SDK are:

**VideoFilter Plugin PB:** This is a video filter, with one video input and one video output. It is essentially the source code for the "Colorizer" actor, without the custom user Interface element (the three bars displaying the color change.)

## **STEP 5: DEBUGGING**

After making your modifications and successfully building your plugin, you can debug it by choosing **Debug > Debug Executable**. This will start up Isadora and begin the debugging process.