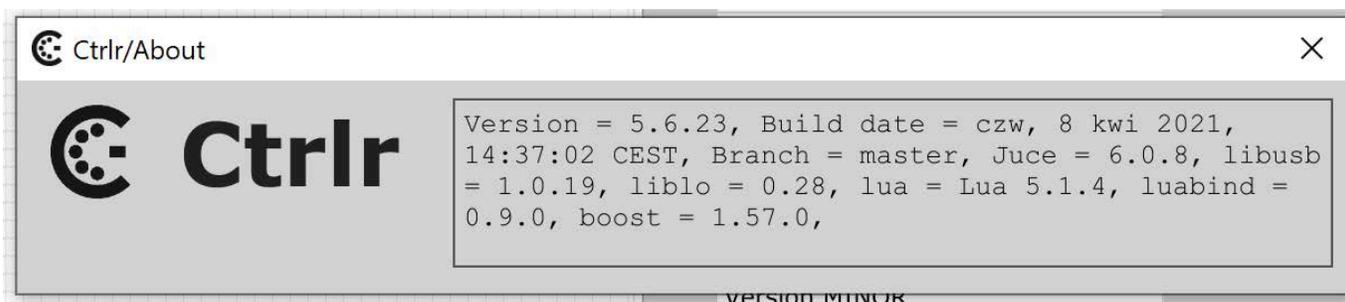


This guide describes the steps I took to compile Ctrlr for macOS with Xcode.

Below is the About Ctrlr window of the compiled version:



The publishing date of this version is: Thursday, April 8, 2021 (I looked up 'czw' and 'kwi'. These are shorts for the day and month in Polish ;-))

The Xcode version I used is v11.3.1
macOS is v10.14.6 Mojave

Before you can download Xcode you need an Apple developers account, which is free, btw.

Download the Ctrlr source files

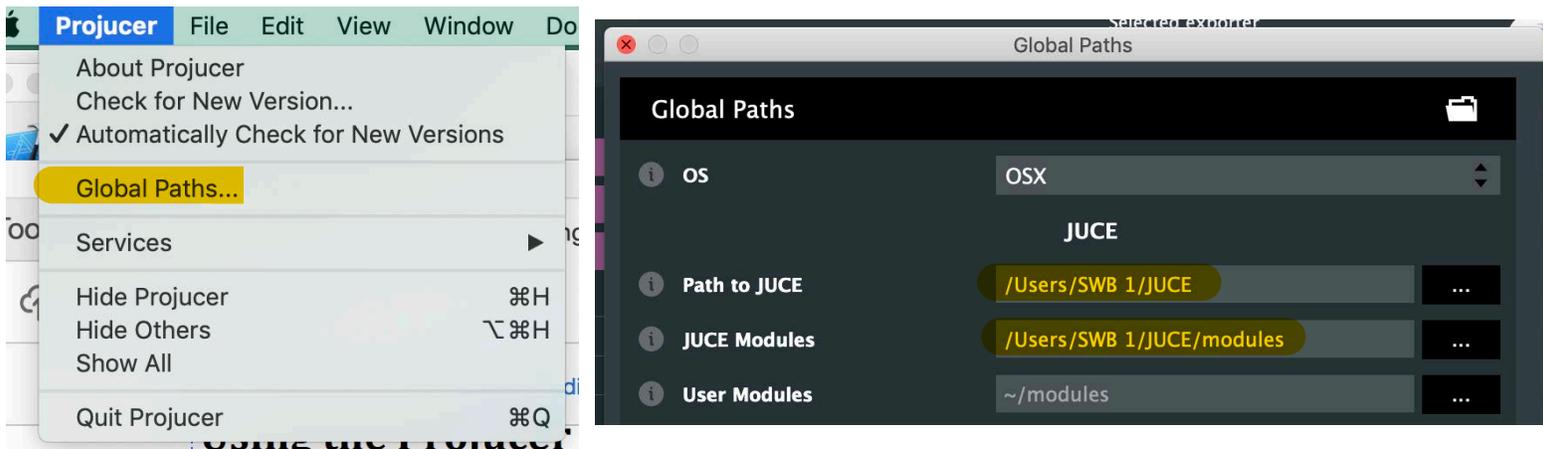
The first thing you have to do of course is download the latest master from the Ctrlr Github. First step you have to do after downloading is unzipping 'boost.zip' You will find this file in ..\ctrlr-master\Source\Misc\boost\. Just unzip it and do nothing further.

Using the Projucer app

Projucer is needed as an intermediate step to load the Ctrlr project properly into Xcode. Projucer comes with JUCE, which you have to download. Select your plan. I choose the free one. After you have downloaded and installed JUCE go to the JUCE folder, where you'll find Projucer.exe.

Building Ctrlr

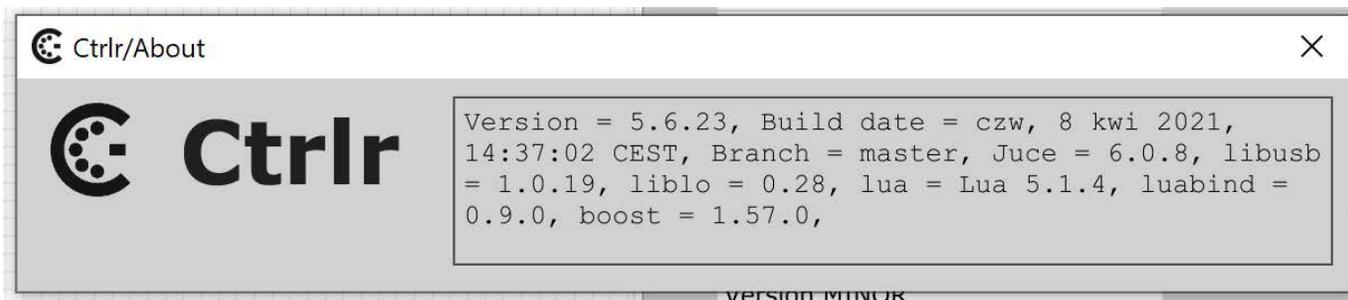
Start Projucer and then check the first two global paths settings. They should point (by default) to the downloaded JUCE folder.



Now choose File>Open... and pick the file: /ctrlr-master/Ctrlr.jucer. This file contains all the (source) files you need to build Ctrlr. Click on the export button of Projucer. Xcode will start and the Ctrlr project is loaded.

This guide describes the steps I took to compile Ctrlr for macOS with Xcode.

Below is the About Ctrlr window of the compiled version:



The publishing date of this version is: Thursday, April 8, 2021 (I looked up 'czw' and 'kwi'. These are shorts for the day and month in Polish ;-))

The Xcode version I used is v11.3.1
macOS is v10.14.6 Mojave

Before you can download Xcode you need an Apple developers account, which is free, btw.

Download the Ctrlr source files

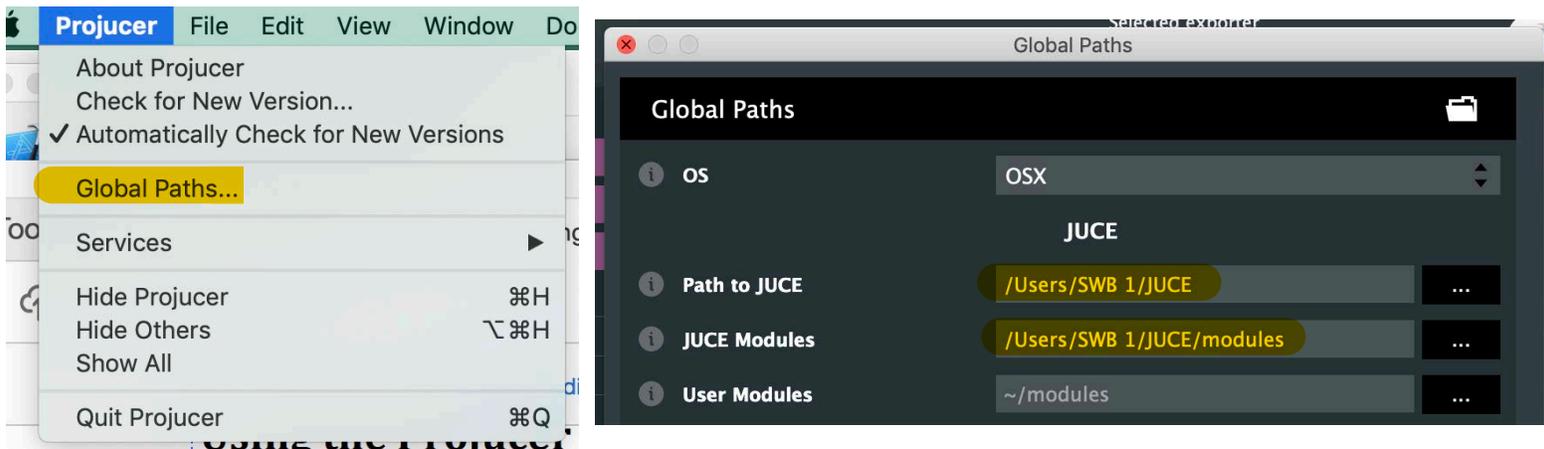
The first thing you have to do of course is download the latest master from the Ctrlr Github. First step you have to do after downloading is unzipping 'boost.zip' You will find this file in ..\ctrlr-master\Source\Misc\boost\. Just unzip it and do nothing further.

Using the Projucer app

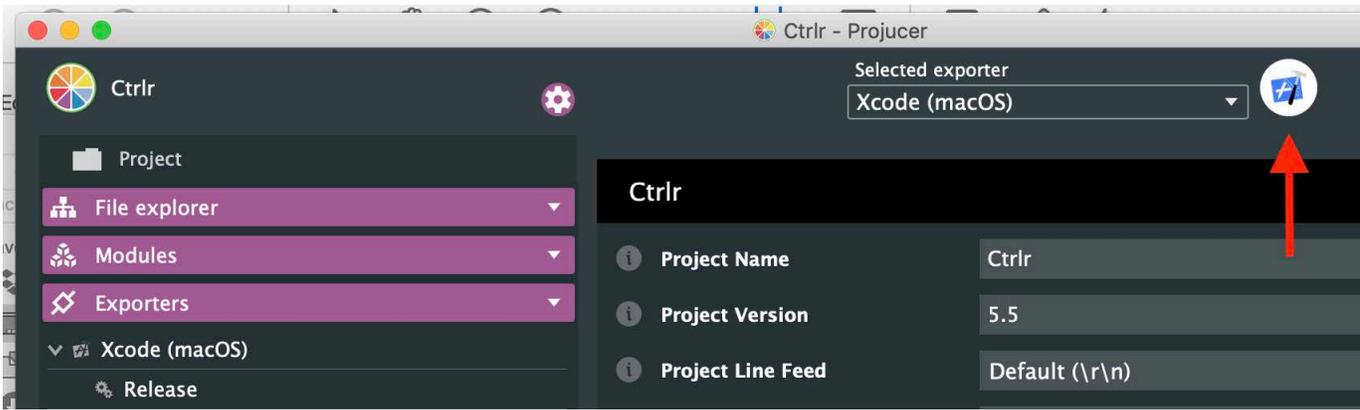
Projucer is needed as an intermediate step to load the Ctrlr project properly into Xcode. Projucer comes with JUCE, which you have to download. Select your plan. I choose the free one. After you have downloaded and installed JUCE go to the JUCE folder, where you'll find Projucer.exe.

Building Ctrlr

Start Projucer and then check the first two global paths settings. They should point (by default) to the downloaded JUCE folder.

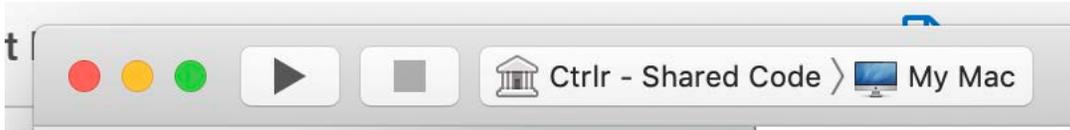


Now choose File>Open... and pick the file: /ctrlr-master/Ctrlr.jucer. This file contains all the (source) files you need to build Ctrlr. Click on the export button of Projucer. Xcode will start and the Ctrlr project is loaded.



Building the Ctrlr_SharedCode

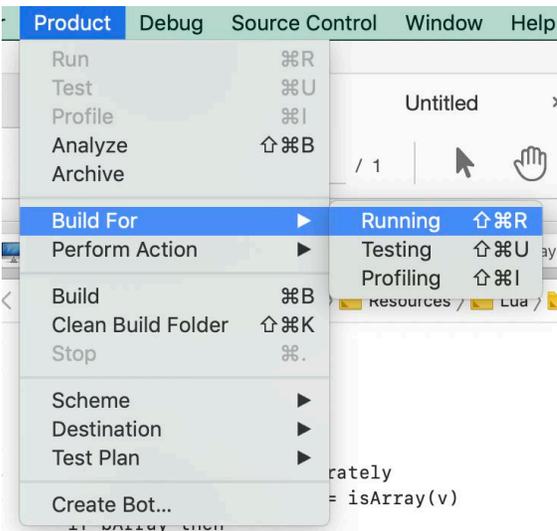
First step in building is to build the Ctrlr - Shared Code. Pick it from the list.



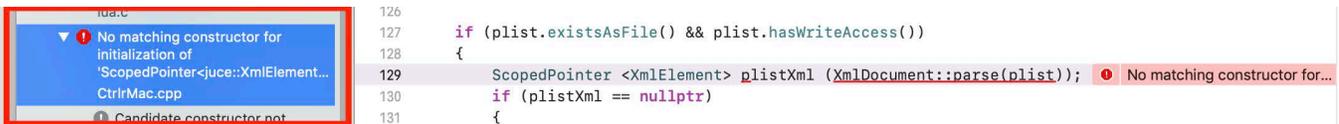
After successful builds, at the end I decided to choose building for 'release'. To accomplish this choose 'Edit Scheme...' and in the window change 'debug' for 'release'.



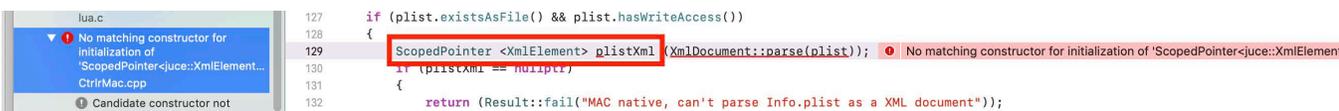
Next start the actual building of Ctrlr - Shared Code:



The build will fail with the message as indicated in the left column:



To solve this error replace the text on line 129 'ScopedPointer' with the text 'std::unique_ptr':



so it looks like this:

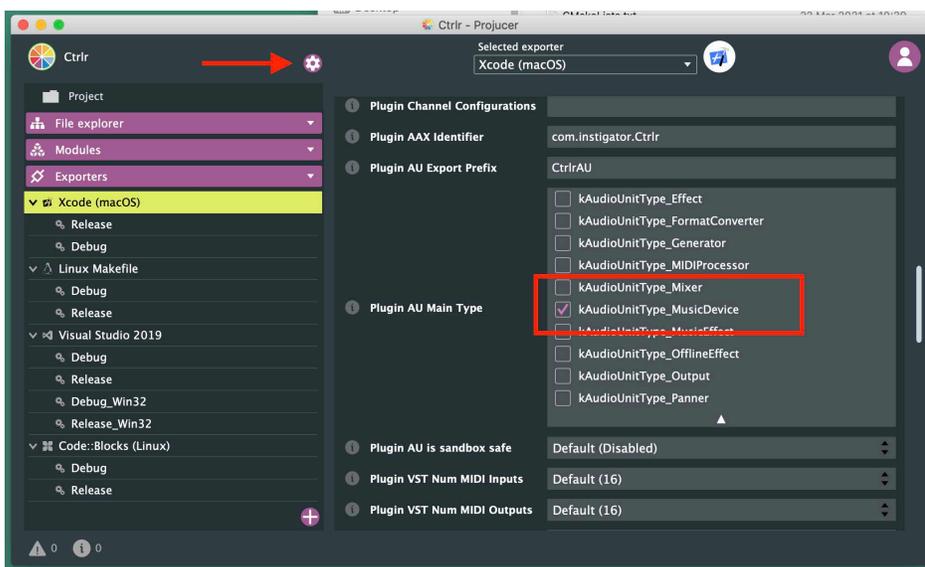
```
129         std::unique_ptr <XmlElement> plistXml (XmlDocument::parse(plist));
```

Xcode replaces automatically all the text ScopedPointer to std::unique_ptr. Wonderful!

After this correction the build succeeds.

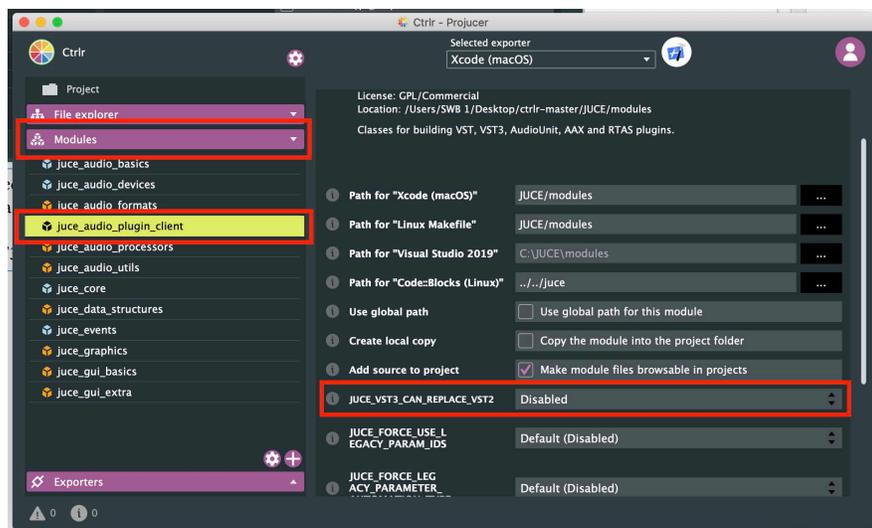
Next build the Ctrlr - Standalone Plugin. Don't forget to set the scheme to release. If all is well, this build also succeeds.

Before you can build the AU plugin, you need to go back to Projucer. In Projucer, in the field 'Plugin AU Main Type', check 'kAudioUnitType_MusicDevice'. To get to this window click on the settings icon (indicated by the red arrow) and scroll down until you find the 'Plugin AU Main Type' settings window. You probably have to expand the window to see the required option:



Next export the project again to Xcode. Xcode starts up and choose building the AU plugin for 'release', by following the steps described earlier in this guide for building Ctrlr. The result is a successful build of the AU plugin.

For building the VST3 plugin, again you need to go back to Projucer to adjust a setting. Click 'Modules', click 'juce_audio_plugin_client' and set 'JUICE_VST3_CAN_REPLACE_VST2' to 'Disabled' and export the project to Xcode:



Xcode start once again. Choose building the VST3 plugin for release and start building. Result is a successful build of the VST3 plugin.

As I did write on the Ctrlr github page, building a VST plugin was unsuccessful. I found out that JUCE does not support this plugin anymore, because Steinberg did deprecate the plugin. When you are able to find the (old) VST SDK and know how to work with it in JUCE or Xcode, then you might be able to build the VST plugin.

Last plugin I did build was the AUv3 AppExtension. For me this format is new, so I don't what to do with it. Maybe you know. If so let me know!

You will find all your builds in: `/ctrlr-master/Builds/MacOSX/build/Release/`

Tip! To test the AU and VST3 plugin, or for that matter every audio plugin on your computer, JUCE comes with the AudioPluginHost.app, but first you have to build this app yourself.

Go to `../JUCE/extras/AudioPluginHost/` and double click the file `AudioPluginHost.jucer`. Projucer will start. Export the project to Xcode and build the app. Start the app and drag the plugins you want to test in the window. Double click the plugin and the content will show. There are a few more options in the app which you certainly will find out by yourself!

This is the end of the guide. Of course you may always contact me with questions, (text) improvements etc. Please do so via the Ctrlr github, so others may also profit!