Compile Ctrlr for Windows

****

Ctrlr is a program made by Roman Kubiak. It's open-source, cross-platform, and able to control your synthesizer/ midi device as a standalone or VST inside almost any DAW. You can download already existing panels or create your own one. The most stable version for windows is Ctrlr 5.3.201. You can download an installable version from: https://ctrlr.org/ No need to compile then.

When to compile? Well, if you are an experienced C++ programmer and want to help fix some bugs or add "JUCE" classes or functionality. Feel free. But that's when you need to compile.

A word of CAUTION. I'm not responsible for f#cking up your Ctrlr installation or your windows setup. After a successful build, windows 10 might see the compiled version as the version to open panelz. Best thing to do then is to assign the bpanelz extension back to the one in the Ctrlr folder of your original installation.

Keep in mind that you might need Microsoft Visual Studio 2010 (Named MVS 2010 from now on .). That is the version that gave me a successful build. I haven't tried MVS 2013 yet. Why not the newer versions of MVS? Because the solution files to make a build easier are not provided for those versions.

When running into problems, don't give up. You might have overlooked something. If you forget one step or forget to set some properties for a certain file, it might rain errors. However, you can always open a topic at the Ctrlr Forum. https://ctrlr.org/forums/

****

Step 1: Get the Ctrlr commit from the Ctrlr repository.

Sounds easy but there are some catches. Go to: https://github.com/RomanKubiak/ctrlr. The Master branch you can download from the start page is Version 6 of Ctrlr. It's not the most stable version. To get version 5.3.201 you need to take the next steps:

-- On the left, there is a combo box with "Master". Change it to "Stable"
-- On the right, you see how many commits there are next to a clock symbol. Click on it.
-- On the bottom of the next page you can go back in time by clicking "older"
-- Go to March 29, 2016, and at the right of that bar, click at the <> symbol.

This is, to my knowledge, the commit for version 5.3.201. Or close to it.

-- Click on the green "Code" button. I would advise to download the zip file.
-- Click on the "Panels @ 61f93ee folder and do the same at the next page.

Now if you go back one page. You are back at the March 29, 2016 commit. Take a look inside the JUCE folder. Notice there is only a modules folder inside it. To compile Ctrlr you need "The Introjucer" which is not provided here. Therefore you need to download the complete JUCE Commit.

-- Go to https://github.com/juce-framework/JUCE , this time click on the tag symbol with a number of tags next to it. Go to tag nr: 4.2.0, dated April 6, 2016, and click it. Download the Source code (zip). So now you should have 3 zip files: Ctrlr, Ctrlr Panels and JUCE.

-- Unpack the Ctrlr Zip file to an easy to find folder. Unpack the "Panels" and "JUCE" zip files to the corresponding folders in the Ctrlr folder.
-- Folder structure for the panels should be: ../Ctrlr/Panels/Demo Panels/*demo panels*
-- Folder structure for JUCE should be: ../Ctrlr/JUCE/*all JUCE folders and files*
-- There is a Boost folder. Open it. There is a boost zip file. Unpack it into the Boost folder. The folder structure should be: ../Ctrlr/Boost/boost/*all unpacked boost files*

****

Step 2: Compile "The Introjucer"

As said, you need "The Introjucer" to debug or compile Ctrlr.
-- Go to: ..Ctrlr/JUCE/extras/IntroJucer/Builds/VisualStudio2010
-- Double click the "The Introjucer.sln" file and MVS 2010 should open.
-- At the top of MVS you see a Combobox with "Debug" or "Release". Set to "Release"
-- Click the green play button and see the magic happen. It should compile "The Introjucer" and put it in a "Release" folder inside the folder in which you found the sln file. Leave it there.

****

Step 3: Compile or Debug "Ctrlr"

Make sure you have the right windows SDK or else the build might fail. Also when installing MVS 2010 make sure you check that you are compiling for C++. However, if you run into unexplainable errors, my advice is to check if those 2 things are done. They came with my installer of MVS 2010. Could differ from yours. In MVS 2010 under "Help" you can see all the installed components.

Mine shows:
****
Installed Version: Ultimate
Microsoft Office Developer Tools   01019-532-2002102-70520
Microsoft Visual Basic 2010   01019-532-2002102-70520
Microsoft Visual C# 2010   01019-532-2002102-70520
Microsoft Visual C++ 2010   01019-532-2002102-70520
Microsoft Visual F# 2010   01019-532-2002102-70520
Microsoft Visual Studio 2010 Architecture and Modeling Tools   01019-532-2002102-70520
Microsoft Visual Studio 2010 Code Analysis Spell Checker   01019-532-2002102-70520
Microsoft Visual Studio 2010 Team Explorer   01019-532-2002102-70520
Microsoft Visual Web Developer 2010   01019-532-2002102-70520
Crystal Reports Templates for Microsoft Visual Studio 2010
Microsoft Visual Studio 2010 SharePoint Developer Tools   10.0.30319
****

-- Go to: ..Ctrlr/Builds/   open the Ctrlr_Release.sln file.
(This is the MVS 2010 solution file to build the Standalone and the VST in one go. If you want to create them separately you can go to the ../Ctrlr/Builds/Generated/Windows folder and choose the one you want.)

To the left, you see a "Solution Explorer" pane. That is where most of the work is done. It now shows: "Solution 'Ctrlr_Release' (2 Projects)" with the 2 projects being: Ctrlr_Plugin_VST and Ctrlr_Standalone. The latter in bold.

In the ribbon above it, you see the green playback symbol with Debug and Win32 next to it. Afaik you can set Debug to Release but you need a Win32 file to produce the X64 one. So leave it at Win32 for now.

-- When I say "Do "All" steps in the same way for the Ctrlr_Standalone", the steps go from here. Needless to say that "Ctrlr_Plugin_Vst needs to be replaced by "Ctrlr_Standalone".

-- At the Solution Explorer, right-click "Ctrlr_Plugin_Vst" and open "Properties".
(Make sure you click the "Ctrlr_Plugin_Vst" with the MVS2010 Icon in front of it,
not the folder that's named Ctrlr_Plugin_Vst". The latter opens a properties pane on the right of the MVS2010 window. The first one opens the Ctrlr_Plugin_VST Properties Pages.)

In the "Ctrlr_Plugin_VST Properties Pages" go to:
-- Configuration Properties/ C/C++/ Precompiled Headers
To the right there is a "blank" field next to "Precompiled Header"
-- Change the Combobox to "Create (/Yc)

Then go to: Configuration Properties/ C/C++/ Command Line
-- Make sure "Inherit from parent or project defaults" is checked
-- At the bottom you see the "Additonal Options" that shows:

/MP /Yu"stdafx.h" /Fp"stdafx.pch" /Zm2000 /bigobj
-- Change /Zm2000 to /Zm200

-- At the Solution Explorer go to:
Solution 'Ctrlr_Release' (2 projects)/ Ctrlr_Plugin_VST/ Ctrlr_Plugin_Vst/ Core
There should be a "stdafx.cpp" file in that folder. Right-click and choose options.
-- Configuration Properties/ C/C++/ Precompiled Headers
To the right, there is a "blank" field next to "Precompiled Header"
-- Change the Combobox to "Create (/Yc)
Then go to: Configuration Properties/ C/C++/ Command Line
-- Make sure "Inherit from parent or project defaults" is checked

-- Do "ALL" those steps in the same way for the "Ctrlr Standalone"

Last But Not Least:
When you try to build it now, it might throw an error/ warning about **"CtrlrRevision.h"**.
This file isn't in the Core folder mentioned above. But there is a "**CtrlrRevision.Template**" in the same folder as the stdafx.cpp file. Rename it to **CtrlrRevision.h**. Or, while you are at it, change the build and revision name inside that file into something to your own liking.

Last:
When debugging or building the solution file, keep an eye on the logs. When there are errors, google the errors. It might give you a push in the right direction. Or else, go to the Ctrlr forums and open a new topic or search for the "Compiling CTRLR on Windows 10 Manual" topic. There's a search function, use it.

When MVS2010 is saying that **symbols are not loaded**, go to the "Debug" menu and choose "Options" and select "Symbols". Check "Microsoft Symbol Servers" and you should be ready to go.

G00d luck

Tedjuh