

Tutorial: creating a Sensomusic Usine user module on Windows

Version 1.0 (20th April 2018): by Benoit Bouchez (alias iModularSynth)

Version 1.1 (20th April 2018): typos corrected.

Version 2.0 (23rd February 2024): Overhaul.

Introduction

This document is a short tutorial explaining how to create "from scratch" a user module for Sensomusic Usine using Visual C++ on Windows platform. It is not meant to explain how Usine modules are working, the SDK User Manual made by Martin Fleurent is perfect for that.

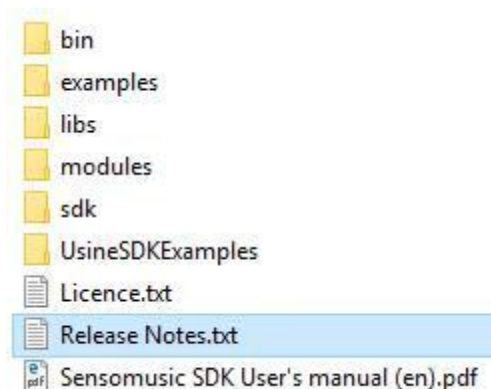
The example given here was created for Usine Hollyhock 6 and should be retro compatible with most Usine Versions since HH2.

Preparing the computer

First, Visual Studio (Express or Developer edition) must be installed on your computer (and you are supposed to know how to use it...).

Download the Usine SDK [BrainModular's website](#) and copy the whole directory on your hard disk. The SDK contains excellent examples to understand how Usine works and what you must implement to make your own modules. We will concentrate here on the steps to follow to create modules from scratch (you can then copy source code from the projects in the SDK to have a good starting point).

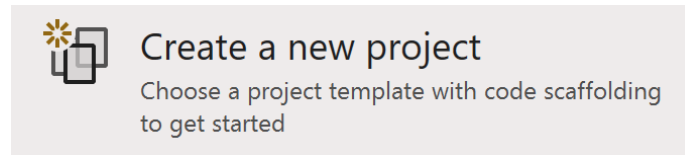
Once the SDK is copied on your hard disk, you should see the following structure in the SDK folder.



Technically, Usine user modules are nothing else than DLL, but you need to define some specific properties in the project to make sure that Usine can load and use them. Screenshots in this tutorial are made under Visual Studio Community 2022. Other versions from Visual Studio may look slightly different but the steps remain the same.

Creating the project in Visual Studio

Start Visual Studio. In File menu, select "Create a new project":

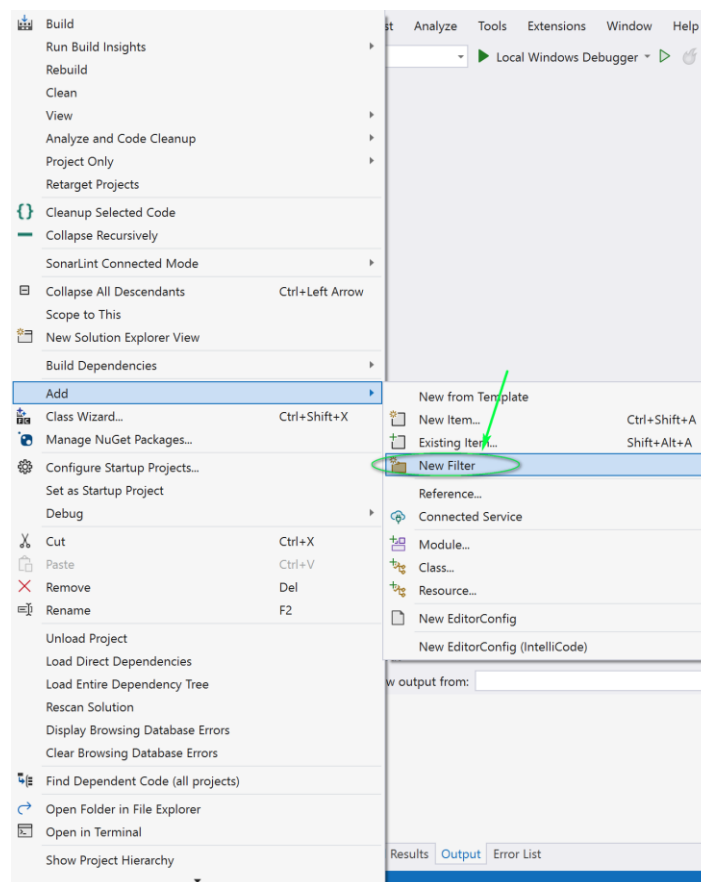


On the next page, select "Empty C++ project" and enter the chosen name:

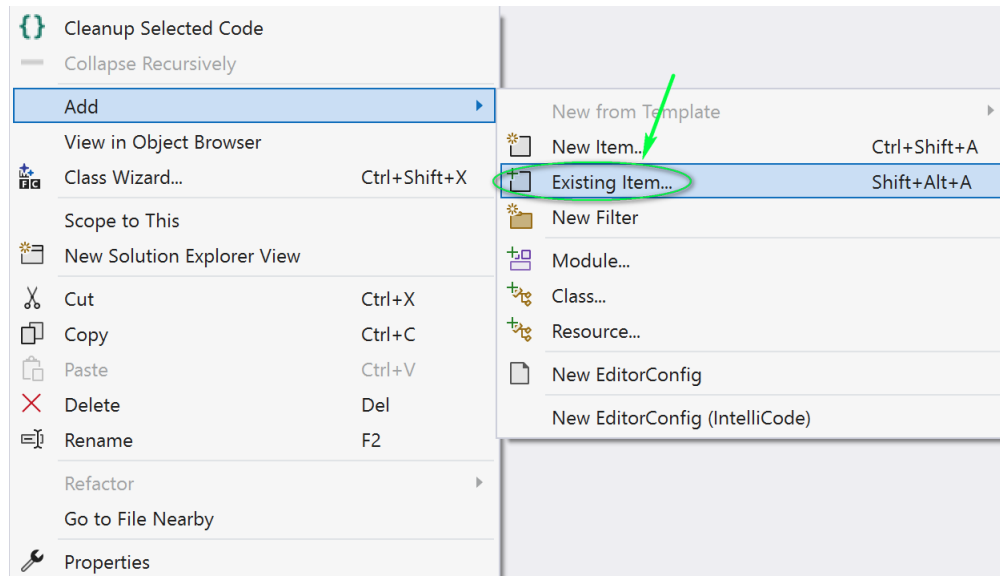


Adding Files to your project

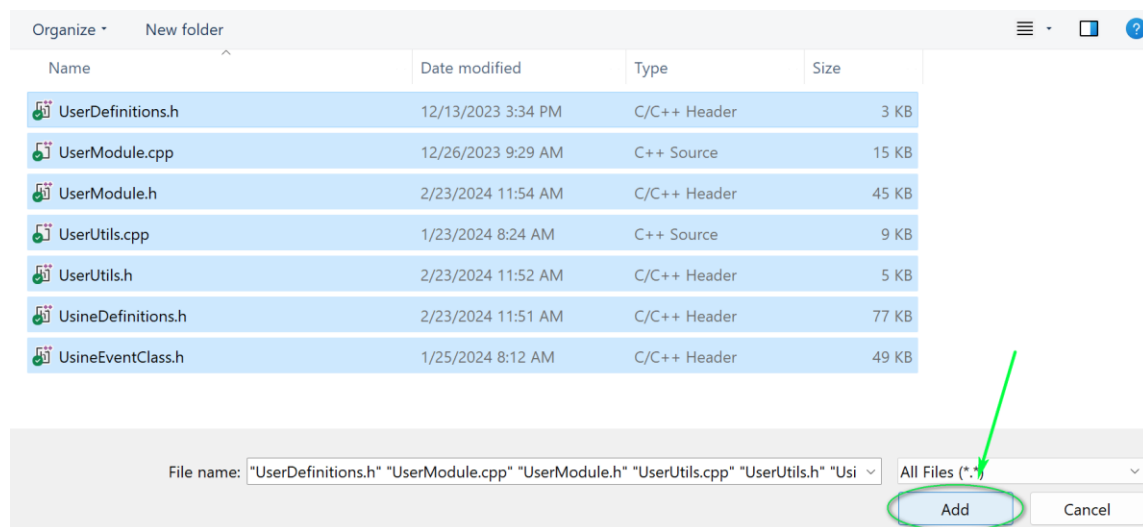
First you should add the SDK files to your project. To do so, right click on the solution name in the solution explorer and navigate to "Add" → "New Filter":



Rename the newly created filter to "SDK" then, right click it and navigate to "Add" → "Existing Item...":

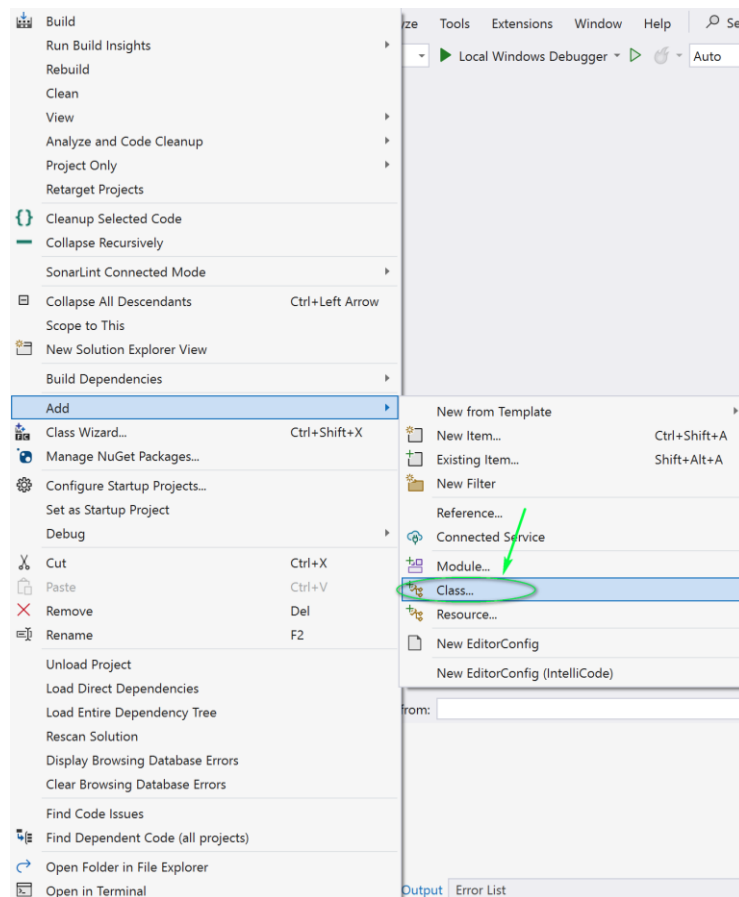


In the following popup screen, find Usine's SDK on your computer, navigate to its "sdk" subfolder, select all contained files and click "Add":

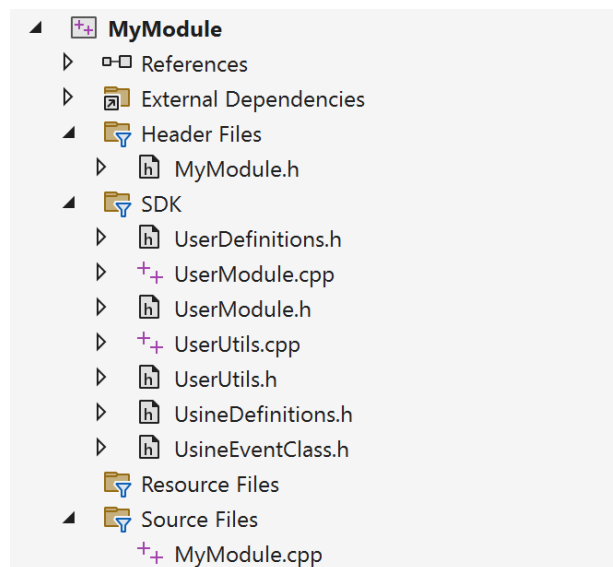


You will at least need one C++ class to build your module. In the future you may freely add more classes to your project.

To add your class to the project, right click on the project's name in the solution explorer and navigate to "Add" → "Class...". On the next popup screen, select the wanted name of the class and click "Ok":

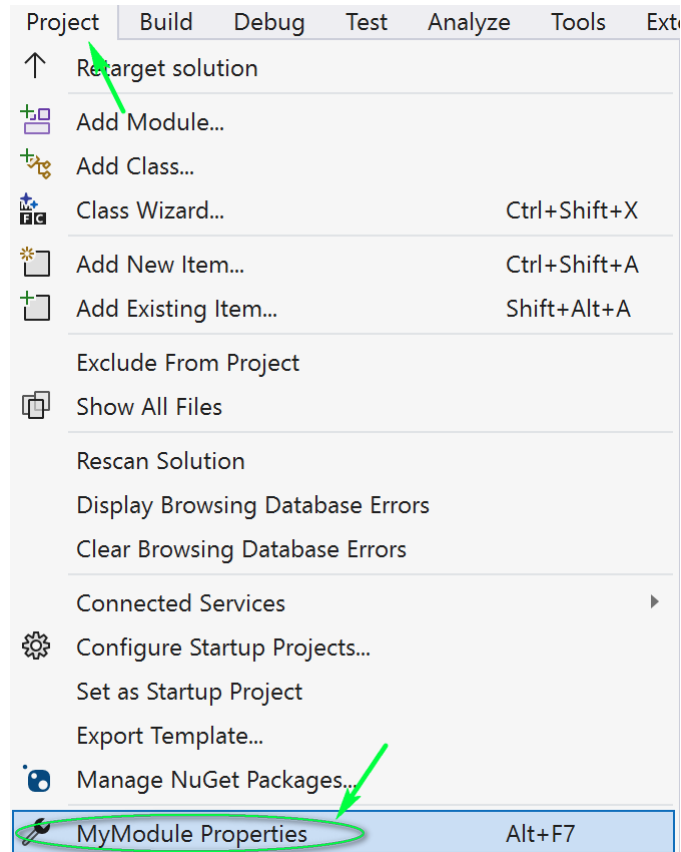


You should now have a project with the following architecture:



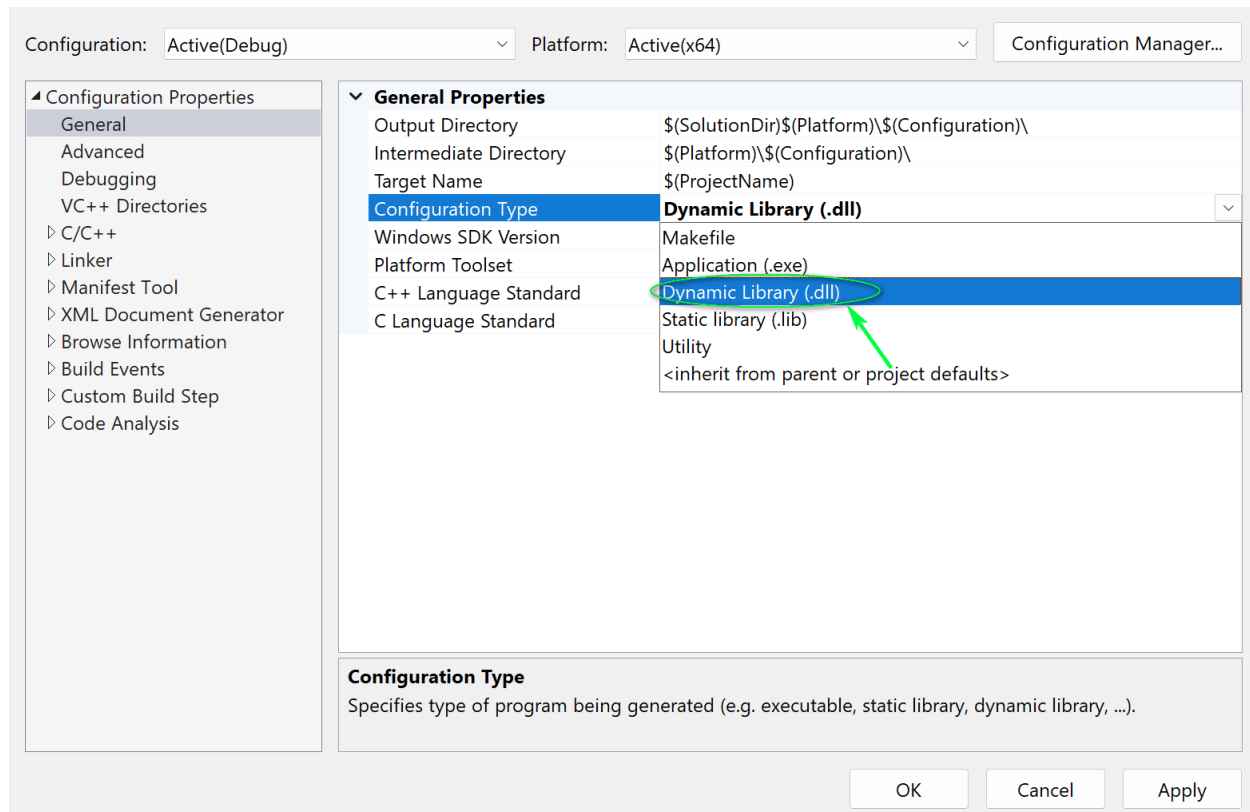
Defining build settings

In this section we will only consider a debug build configuration for Intel x86_64, the steps taken here must be repeated for any other build configuration you may choose. First open the project's properties by opening the "Project" tab in the top toolbar then "Properties":



This will open the project's properties window. This window contains multiple sections, we will go over what settings need changing for each of them next.

First in the general tab change the Configuration Type to "Dynamic Library (.dll)". You may also need to change the C++ Language Standard, Usine's SDK should support any standard for C++11 onwards. As this setting depends on what you're doing we will leave it as default:



Then navigate to the "Advanced" tab, here you'll need to change two settings:

Target File Extension **.usr-win64**
 Character Set **Use Multi-Byte Character Set**

In the "C/C++" → "General" tab, add the path to the sdk folder on your computer, this path can either be absolute or relative to the solution:

Additional Include Directories **..\..\sdk**

We've covered every step needed to build a user module for Usine, you may want to set post build events to copy your binary into Usine's installation folder automatically.

Debugging your module

To use Visual Studio's debugger, you need to declare the executable used to start your module (don't forget that a user module in Usine is nothing than a DLL, it can't then run by itself)

In your project's properties, go to the Debugging tab. Click on Command Line editing area and when the drop-down list appears (down arrow on the right), scroll to "Browse". Browse to the Usine executable located into the bin folder of Usine (the executable located at the root of Usine's installation folder is a dummy .exe file that only serves to launch the real program in the bin directory):

