

# Using cmake for building VST 3 plug-ins

This tutorial explains how to use *cmake* with **VST 3 SDK**.

## CMake for building VST 3 plug-ins

The SDK provides a set of cmake files allowing you to compile the included samples and to develop new plug-ins.

- Download cmake from: <https://cmake.org> or use a package manager for your OS (See [How to set up my system for VST 3](#)).
- You can use the command line or the cmake editor ([cmake-gui](#)).

### On this page:

- [CMake for building VST 3 plug-ins](#)
- [Command line for Windows](#)
- [Command line for macOS](#)
- [On Linux with QtCreator](#)
- [Use of cmake-gui](#)
- [Available SMTG cmake options](#)
- [Using your IDE for compiling the examples](#)

### Related pages:

- [How to set up my system for VST 3](#)
- [Building the examples included in the SDK](#)

## Command line for Windows

Example for building Microsoft "Visual 2019" solution:

```
// go in to the folder where you extracted the VST 3 SDK
mkdir build
cd build
cmake.exe -G "Visual Studio 16 2019" -A x64 "..\VST3_SDK"
// or without symbolic links
cmake.exe -G "Visual Studio 16 2019" -A x64 "..\VST3_SDK" -SMTG_CREATE_PLUGIN_LINK=0

// note: you can find the string definition for different Visual Studio Generators in the cmake online
documentation
```

## Command line for macOS

Example for building Xcode project:

```
// go in to the folder where you extracted the VST 3 SDK
mkdir build
cd build
/Applications/CMake.app/Content/bin/cmake -G"Xcode" "../VST3_SDK"
```

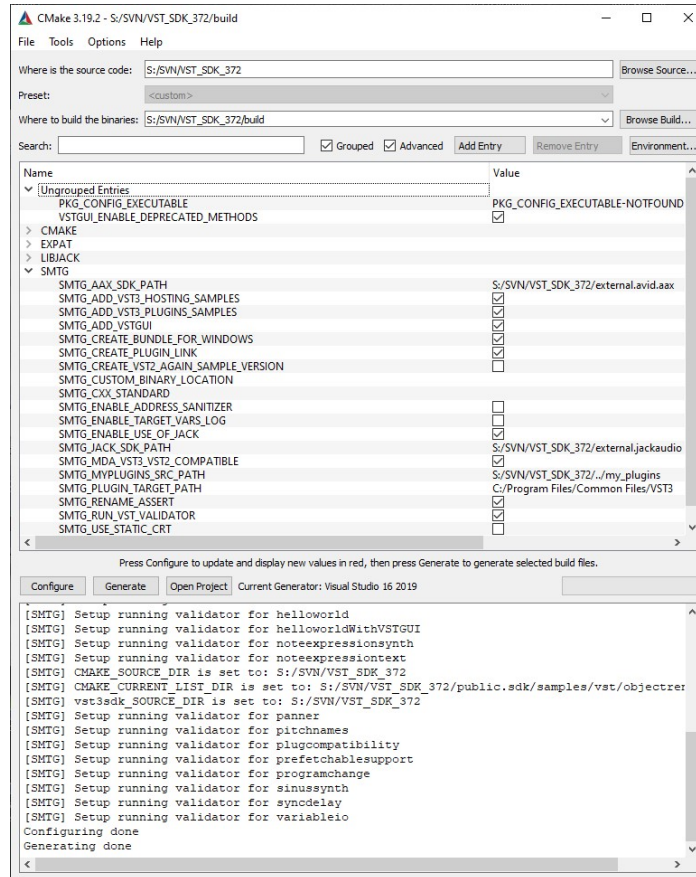
## On Linux with QtCreator

You can use QtCreator 2.3.1 (or higher)

```
start QtCreator 2.3.2
open the CMakeLists.txt located at the top of the VST 3 SDK folder
click on the menu Build->Run CMake
```

## Use of cmake-gui

start the CMake (cmake-gui) application  
 set "Where is the source code" to the location of the "VST3\_SDK" folder  
 set "Where to build the binaries" to a build folder of your choice  
 click on "Configure"  
 click on "Generate" for creating project/solution



Example of cmakegui application on Windows

- Compile with cmake command line

```
cd build
cmake --build
```

- Choose a specific compiler with cmake (command line on Linux)

```
cmake -DCMAKE_C_COMPILER=/usr/bin/clang -DCMAKE_CXX_COMPILER=/usr/bin/clang++

or

cmake -DCMAKE_C_COMPILER=/usr/bin/gcc -DCMAKE_CXX_COMPILER=/usr/bin/g++
```

## Available SMTG cmake options

- **SMTG\_AAX\_SDK\_PATH**: Here you can define where the AAX SDK is located (if needed)
- **SMTG\_ADD\_VST3\_HOSTING\_SAMPLES**: Add VST3 hosting samples to the solution (default ON)
- **SMTG\_ADD\_VST3\_PLUGINS\_SAMPLES**: Add VST3 plug-in samples to the project (default ON)
- **SMTG\_ADD\_VSTGUI**: Add VSTGUI support (default ON)
- **SMTG\_BUILD\_UNIVERSAL\_BINARY**: Build universal binary (32 & 64 bit) (Mac only)
- **SMTG\_COREAUDIO\_SDK\_PATH**: Here you can define where the COREAUDIO SDK is located (Mac only, if needed)

- **SMTG\_CREATE\_BUNDLE\_FOR\_WINDOWS:** Create bundle on Windows for the [VST3](#) plug-ins (new since 3.6.10! Windows only) (default ON)
- **SMTG\_CREATE\_PLUGIN\_LINK:** Create symbolic link for each [VST3](#) plug-in in `#{VST3_FOLDER_NAME}` folder (you need to have Administrator rights on Windows or change the Local Group Policy to allow the creation of symbolic links) (default ON)
- **SMTG\_CREATE\_VST2\_AGAIN\_SAMPLE\_VERSION:** Allows you to create the VST2 version of the Sample Plug-in AGain, be sure that you have copied the VST2 interfaces into the folder `VST_SDK/VST3_SDK/plugininterfaces/vst2.x` (default OFF)
- **SMTG\_CUSTOM\_BINARY\_LOCATION:** Customize output location for binaries
- **SMTG\_CXX\_STANDARD:** C++ standard version used for plugins: 14, 17, 20
- **SMTG\_ENABLE\_ADDRESS\_SANITIZER:** Enable Address Sanitizer
- **SMTG\_ENABLE\_TARGET\_VARS\_LOG:** Enables to log target variables for debugging (new since 3.6.11!) (default OFF)
- **SMTG\_ENABLE\_USE\_OF\_JACK:** Allows you to create the audiohost application using Jack (default OFF)
- **SMTG\_MDA\_VST3\_VST2\_COMPATIBLE:** Build the MDA examples as a replacement for their VST2 counterpart (default ON)
- **SMTG\_IOS\_DEVELOPMENT\_TEAM:** Needed for building the InterAppAudio and AUv3 examples for iOS (Mac only)
- **SMTG\_MYPLUGINS\_SRC\_PATH:** Here you can add your VST3 plug-ins folder
- **SMTG\_PLUGIN\_TARGET\_PATH:** Here you can redefine the VST3 plug-ins folder
- **SMTG\_RENAME\_ASSERT:** Rename ASSERT to SMTG\_ASSERT to avoid conflicts with 3rd party libraries (default ON)
- **SMTG\_RUN\_VST\_VALIDATOR:** Run the VST validator on VST3 plug-ins each time they are built (default ON)
- **SMTG\_USE\_STATIC\_CRT:** Use static CRuntime on Windows (option /MT) (default OFF) (Windows only)

### Preparation on Windows

Generated VST3 Microsoft Visual Studio Projects using the [cmake](#) included in the SDK will create by default symbolic links for each built plug-in in the [official VST3 folder](#), in order to allow this on Windows you have to adapt the Group Policy of Windows. See [Here!](#)

If you do not want to create this link, call [cmake](#) with this parameter:

```
-SMTG_CREATE_PLUGIN_LINK=0
```

## Using your IDE for compiling the examples

- solution/project (`vstsdk.sln/vstsdk.xcodeproj`) is generated in the "build" folder.
- the created plug-ins are located in the "build" folder, in sub-folders ***VST3/Release*** or ***VST3/Debug***.
- In order to allow a DAW to find these plug-ins you have to create links from the [official VST3 Locations](#) to them.