# Creating a plug-in from the Helloworld template

This tutorial explains how to create a new audio plug-in from the Helloworld template included in the **VST 3 SDK**.

**On this page:**

**Related pages:**

- Generate a new plug-in with Project Generator

## Part 1: Getting and installing the VST 3 SDK

For downloading the SDK, see the section " How to set up my system for VST 3".

You have the following possibilities to start a new project:

- You can use the **helloworld** template included in the **VST SDK** and duplicate the folder into a new folder. Adapt each file where the comment mentions it.
- Or, which is **easier** and **recommended**, you can use the **VST3 Project Generator** application included in the **VST SDK.** The following steps show how to use it.

## Part 2: Using the **helloworld** template

The SDK provides a HelloWorld example which you can use to create a new **VST 3 plug-in**:

- Just copy the folder **VST_SDK/my_plugins** containing the HelloWorld example into your development folder.
  - For example: copy **VST_SDK/my_plugins** to D:/Users/Me/Desktop/development/my_plugins

- Now you have to indicate to **cmake** to add this new location to the projects. There are 3 possibilities:
  - Search in **VST3_SDK/CMakeLists.txt** for the comment "# Here you can add your VST3 Plug-ins folder" and specify the path to the folder, for example:
    - ***set(SMTG_MYPLUGINS_SRC_PATH "D:/Users/Me/Desktop/development/my_plugins")***
  - Or when using the **CMake GUI App**, you can specify the new location by using the browser for the variable **SMTG_MYPLUGINS_SRC_PATH**.
  - Or call **cmake** with the option
    - ***-DSMTG_MYPLUGINS_SRC_PATH=D:/Users/Me/Desktop/development/my_plugins***

- You can duplicate the helloworld folder for your plug-in, for example:
  - copy ***D:/Users/Me/Desktop/development/my_plugins/helloworld*** to ***D:/Users/Me/Desktop/development/my_plugins/MyDelayPlugin***

- Adapt the **CMakeLists.txt** files:
  - Open the plug-in **CMakeLists.txt** file with a text editor: ***D:/Users/Me/Desktop/development/my_plugins/MyDelayPlugin/CMakeLists.txt***
  - Change the target name:
    - ***set(target helloworld)*** => ***set(target MyDelay)***
  - Open the folder CMakeLists.txt file located in my_plugins with a text editor in order to add your plug-in to the project:
    - D:/Users/Me/Desktop/development/my_plugins/CMakeLists.txt
  - Add this entry (your newly created folder):
    - ***add_subdirectory(MyDelayPlugin)***

- Generate the project by using the command line or the cmake editor (cmake-gui) as described here: How to use cmake for building VST 3 plug-ins. Your new plug-in should appear in the project afterwards.

- Now you have to adapt some ***uids*** and naming to make your plug-in unique (and not a duplicate of helloworld!)
    1. Rename all strings for your plug-in from ***HelloWorld*** to ***MyDelay*** for example:
        - ***HelloWorldProcessor::HelloWorldProcessor*** to ***MyDelayProcessor::MyDelayProcessor***
    2. Open the file ***MyDelayPlugin/include/plugids.h*** and create new ***uids*** for processor and for controller: you can use GUID creator tools like https://www.guidgenerator.com/ :
        - static const FUID MyProcessorUID (0x2A0CC26C, 0xBF88964C, 0xB0BFFCB0, 0x554AF523);
        - static const FUID MyControllerUID (0xB9DBBD64, 0xF7C40A4C, 0x9C8BFB33, 0x8761E244);
    3. Open the file ***version.h*** and adapt the strings like this:

```
#define stringPluginName "My First Delay"
#define stringOriginalFilename "MyDelay.vst3"
```

4. Adapt ***my_plugins/MyDelayPlugin/resource/info.plist*** by renaming:

```
<string>helloworld</string> => <string>mydelay</string>
<string>com.steinberg.vst3.helloworld</string> => <string>com.steinberg.vst3.mydelay</string>
```

- Now you can start to code for your effect/instrument (see Generate a new plug-in with Project Generator for a step-by-step explanation)
    1. Add parameters in plugcontroller.cpp
    2. Adapt your process algorithm in plugprocessor.cpp
    3. Add persistence in plugprocessor.cpp
    4. Add UI (check SDK examples using VSTGUI)

- Happy coding!