

HowTo 5 – using CMake and an IDE for GSE

A guide to build a Guiliani Streaming Editor (GSE) project using CMake

Product:	Guiliani Streaming Editor (GSE)
Release version:	2.2
Release date:	August 31, 2018

Disclaimer

The information in this document is subject to change without notice and does not represent a commitment on the part of TES Electronic Solutions GmbH. The software described by this document is subject to a License Agreement and may not be copied to other media. No part of this publication may be copied, reproduced or otherwise transmitted or recorded, for any purpose, without prior written permission by TES Electronic Solutions GmbH.

“Guiliani.de”, “Guiliani”, “Guiliani Streaming Editor”, “GSE” and associated logos are (registered) trademarks of TES Electronic Solutions GmbH.

Windows, Visual Studio and Visual C++ are registered trademarks of Microsoft Corporation in the United States and/or other countries.

All other trademarks are the property of their respective owners and use of them does not imply any affiliation with or endorsement by them.

Guiliani, guiliani.de and GSE are products of

TES Electronic Solutions GmbH
Hanauer Landstrasse 328-330
60314 Frankfurt am Main
Germany

Email: support@guiliani.de
Website: <http://www.guiliani.de>
Forum: <https://www.guiliani.de/forum/>
HelpDesk: <https://guiliani.on.spiceworks.com/portal>

Table of contents

1.	Introduction	5
1.1.	Assumed knowledge	5
1.2.	Prerequisites	5
1.3.	Documentation conventions	5
1.4.	Shortcuts	6
1.5.	This document's goal	6
2.	Compile/Re-compile GSE / your project	7
2.1.	Step 1: Using CMake to create a VisualStudio-solution	7
2.2.	Step 2: IDE - Microsoft Visual Studio as an example	10
2.2.1.	Start your solution	10
2.2.2.	Build your solution	13
2.2.3.	Start the GSE out of Microsoft Visual Studio	14
2.2.4.	Include your GSE project to Microsoft Visual Studio	15
2.3.	Step 3: How to continue?	18
2.3.1.	Continuing HowTos	18
3.	Index	19

Table of figures

Fig. 1 CMake: enter source path	7
Fig. 2 CMake: enter build path.....	7
Fig. 3 CMake: new source path.....	8
Fig. 4 CMake: toolchain selection.....	8
Fig. 5 CMake: configuring done	9
Fig. 6 GSE.sln	10
Fig. 7 MS VS 2015 started	11
Fig. 8 Options	11
Fig. 9 Set Color theme.....	12
Fig. 10 GSE as StartUp Project	13
Fig. 11 Rebuild Solution	13
Fig. 12 Successful build	14
Fig. 13 Start Debugging	14
Fig. 14 Set StreamRuntime as StartUp Project and select Properties	15
Fig. 15 StreamRuntime Output Directory	16
Fig. 16 StreamRuntime Working Directory	16
Fig. 17 Set StreamRuntime Working Directory	17
Fig. 18 Run your project	17

1. Introduction

This document explains how to build the GSE and your own project using CMake and an IDE.

1.1. Assumed knowledge

- Basic handling of GSE

1.2. Prerequisites

- Unpacked Guiliani SDK including GSE
- Installed CMake
- Installed Microsoft Visual Studio 2015

1.3. Documentation conventions

Whenever you can use keys from your computer's keyboard, these will be displayed in square brackets (e.g., "To run your project press [Ctrl] + [r].").

Menu commands or file path used in this document will be shown in *italic*.

Text that appears in the software on controls will be printed in **bold and blue**.

▶ Whenever the reader of this document has to do something in his project, the text will start with this triangle.

➡ Results will be shown using this arrow.

In this document, we use icons whenever we will warn the user or will give him additional or important information.



The speech bubble icon will show additional helpful information.



Whenever a text begins with an exclamation mark icon, it contains important information that is essential for the current chapter.



A warning sign icon signals serious issues and potential risks that require your full attention.

1.4. Shortcuts

In the documents, we often select a command from the window. These can be selected by the following short cuts, too:

<i>File</i> → <i>New Project...</i>	[Ctrl] + [Shift] + [n]
<i>File</i> → <i>New Dialog...</i>	[Ctrl] + [n]
<i>File</i> → <i>Save Project...</i>	[Ctrl] + [s]
<i>File</i> → <i>Run Simulation...</i>	[Ctrl] + [r]

1.5. This document's goal

At the end of this How To you will have learned

- how to use CMake for developing your GSE project
- when you have to compile your project/the GSE
- and more

2. Compile/Re-compile GSE / your project

2.1. Step 1: Using CMake to create a VisualStudio-solution



In order to be able to use the GSE and the StreamRuntime / your project after making code changes, it needs to be recompiled with your code present in the build process.

CMake is used to configure the build environment, making the adding of your code very simple once the project has been set up.



The GSE and the StreamRuntime / your project are managed in one CMake project.

To use CMake, please follow the next steps:

- ▶ Download the cmake-gui from <http://www.cmake.org/> and install it.
- ▶ Run CMake and open the main-folder of the Guiliani-SDK as the “Source-folder” in cmake-gui.



Fig. 1 CMake: enter source path

- ▶ Select a new directory as the “Build” directory. For our tutorial, we will create the folder “Build_GSE_2015_HowTo” in the root directory as the build directory.



This folder will contain files generated by CMAKE as well as the resulting binaries.

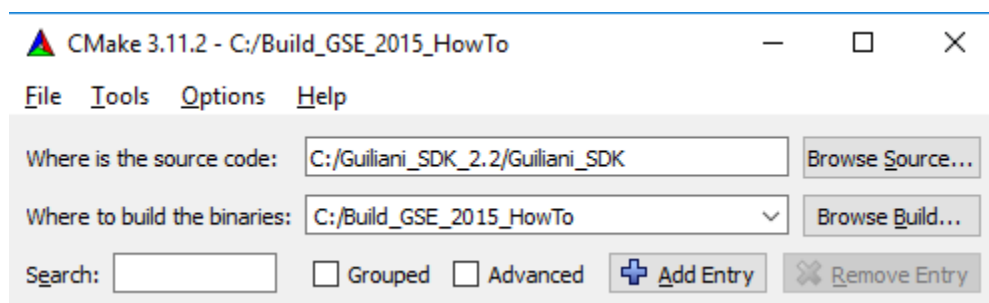


Fig. 2 CMake: enter build path



For this HowTo, we will use the Guiliani SDK for Microsoft VS 2015 and therefore we create and use the folder “Build_GSE_2015_HowTo”



As the source directory is used in a CMake-generated #define-statement, the path must be preprocessor-safe, i.e. it must not contain special characters like "#".



If you do not like long path names, just move the contents of the Guiliani-SDK-folder (i.e. CMake, Guiliani, GSE etc.) to a new one. We have done it and renamed “Guiliani_2.2_SDK_including_GSE_Windows-Desktop_VS2015” to “Guiliani_HowTos_2015”.

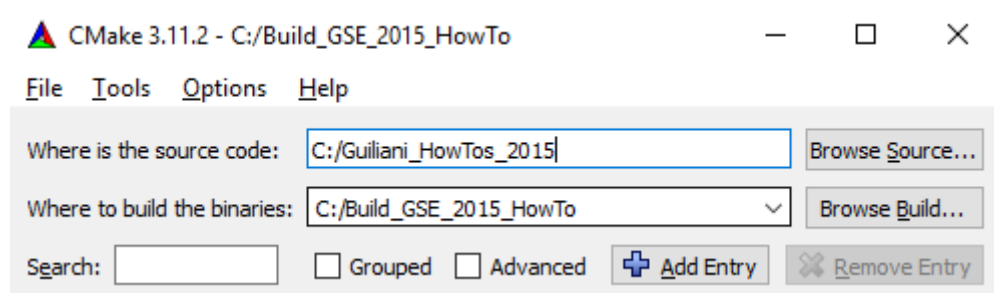


Fig. 3 CMake: new source path

- ▶ Click onto “[Configure](#)”. If the build-folder does not exist CMake will ask you to create it. After that the dialog for selecting the generator will be shown.
- ▶ Here please choose Visual Studio 14 2015.



Do not make any other changes.

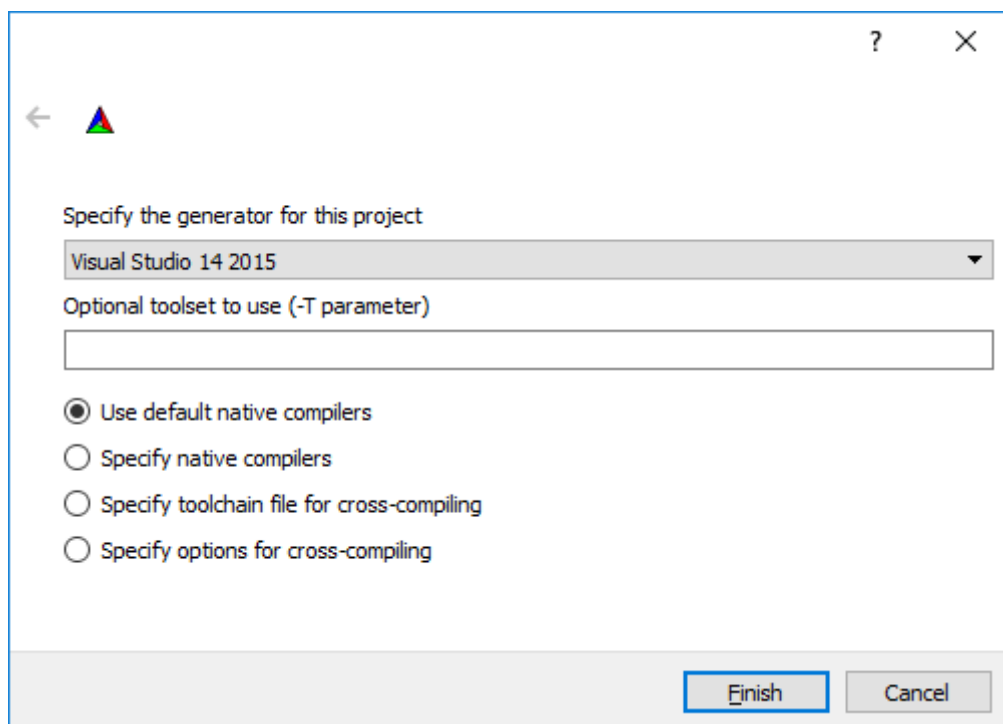


Fig. 4 CMake: toolchain selection



For this HowTo, we will use a Guiliani SDK for Microsoft VS 2015 and therefore we use “Visual Studio 14 2015” as the generator (see above). If you want to use any other build system, please note that the Guiliani-SDK needs to include the libraries for it.

▶ Click onto “**Finish**”.

➔ The project will be configured.



There will be a lot of text at the bottom, mostly in red. Don't worry. That's normal. Only be sure that the text will end with “Configuring done”.

```
GUILIANI_ADD_OGL_PLATFORM has been activated by the application cmake project!  
GUILIANI_ADD_GL_PLATFORM has been activated due to GUILIANI_SET_OGLES_PLATFORM or GUILIANI_SET_OGL_PLATFORM is set!  
GUILIANI_ADD_IMAGE_DECODER has been activated due to GUILIANI_ADD_GL_PLATFORM is set!  
GUILIANI_USE_RTTI has been activated by the application cmake project!  
Configuring done
```

Fig. 5 CMake: configuring done

▶ Click the button “**Configure**” to let CMake accept the preset default values for the project's variables.



These variables need only be adjusted for specific build scenarios, e.g. when a release build is to be made to evaluate performance on an embedded device. So it is of no interest for our “HowTo” project.

➔ Now, there should be no more red text in the upper part of the CMake application.

▶ A click onto the button “**Generate**” will create the platform-specific build files: a “.sln” file and its dependencies.

➔ In the lower left corner, beneath “Configuring done”, there should be written “Generating done”.



The “Configure” and “Generate” steps need to be repeated every time new files are added to the source-directories contained inside the “StreamRuntime”-folder. CMake will then detect new files and modify the build project accordingly.

2.2. Step 2: IDE - Microsoft Visual Studio as an example

To show you how you have to link CMake and the GSE together with your IDE, we will use Microsoft's Visual Studio as an example.

If you are using a Windows PC, you may install Microsoft Visual Studio. You can download a free version of Visual Studio Express 2015 from the internet.



Please be sure, to download the correct Guiliani SDK for your MS Visual Studio version.

2.2.1. Start your solution

- ▶ Open our project's build folder "Build_GSE_HowTo" we created in the previous step. Inside you will find the Solution-file ending in .sln. Double click onto it.

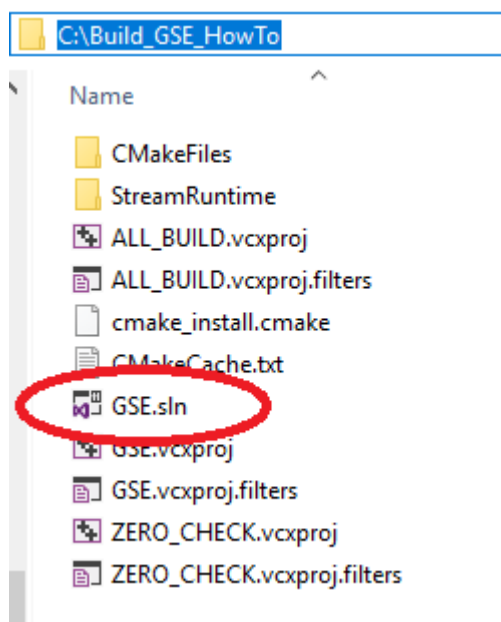


Fig. 6 GSE.sln



If you can see only a file named "GSE" without ending, you should change the Window's settings for folders and allow showing of file-extensions.

➔ The appropriate Microsoft Visual Studio version will be started and the GSE loaded.

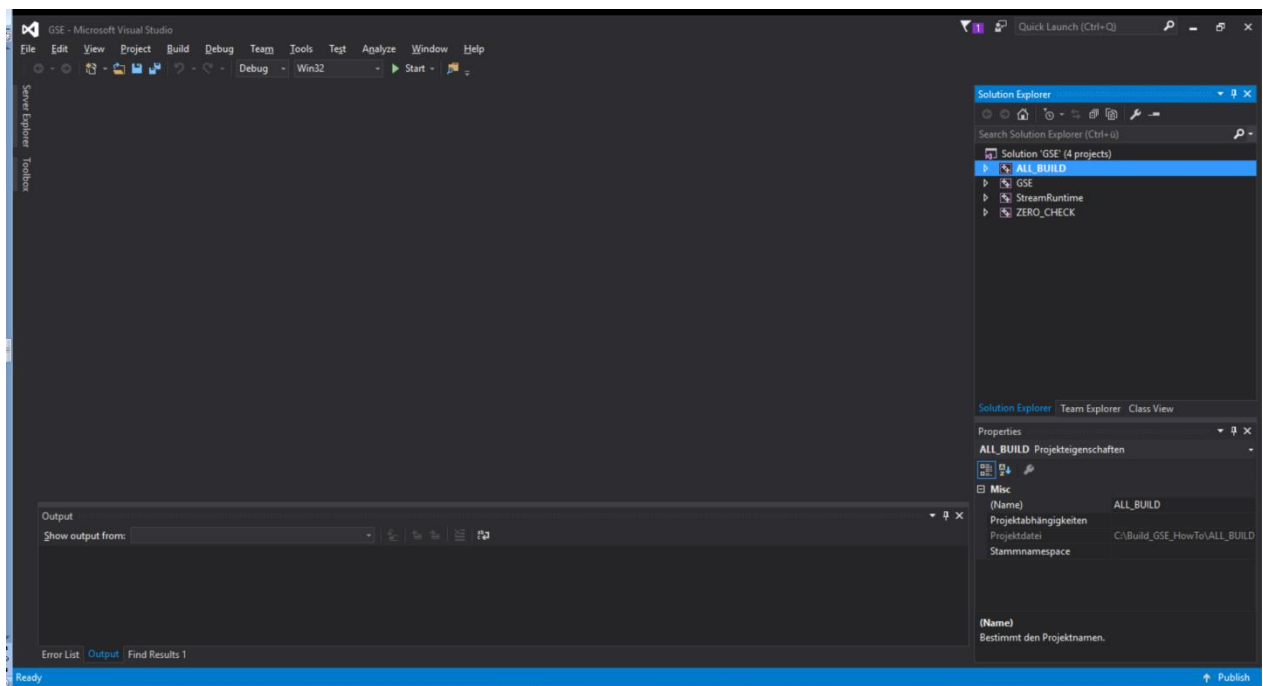


Fig. 7 MS VS 2015 started



If you wonder, why our Visual Studio has black letters on white background: we have changed the Visual Studio's color theme to dark.

You can do this likewise:

- ▶ Choose "Options" from the "Tools" menu.

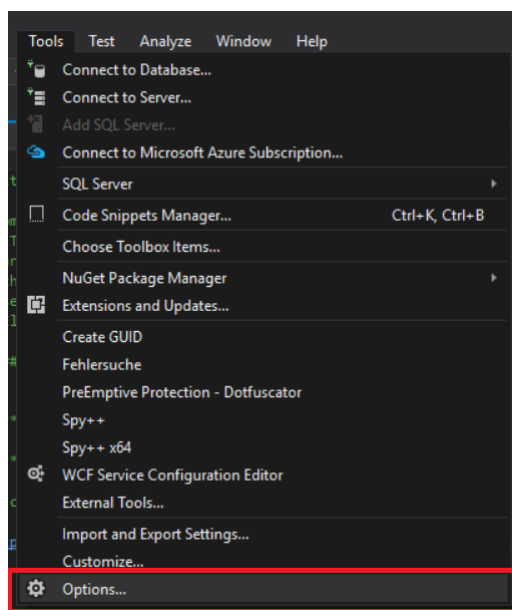


Fig. 8 Options

- ▶ Click onto “General” and change “Color theme” to “dark” (or whatever you prefer).

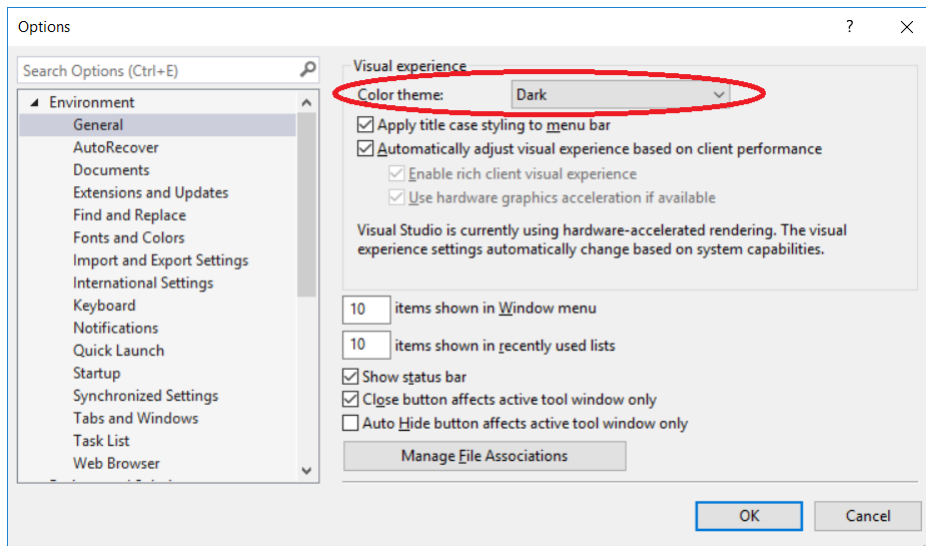


Fig. 9 Set Color theme

But let us start with compilation of GSE.

2.2.2. Build your solution

- ▶ First, set GSE as the StartUp Project.

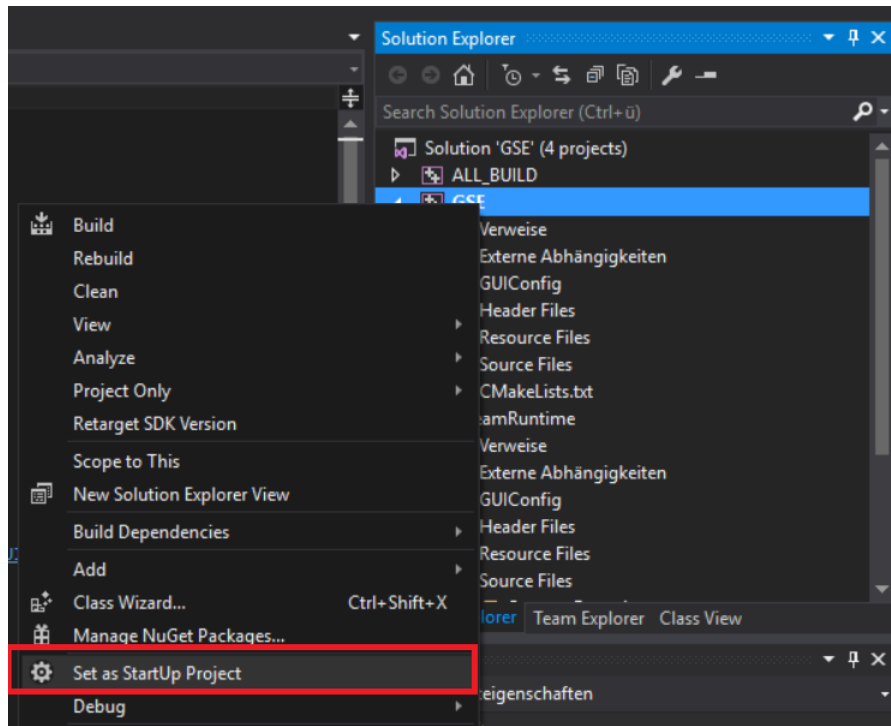


Fig. 10 GSE as StartUp Project

- ▶ From the menu “Build” choose “Rebuild Solution” to make a new compilation of the GSE.

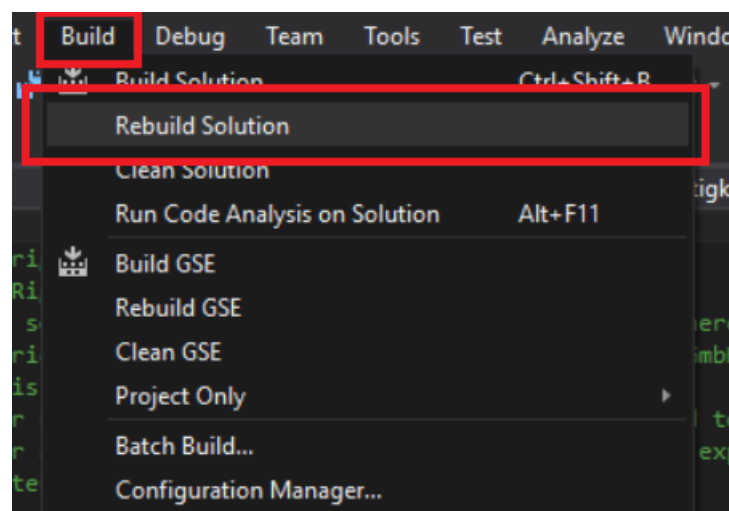
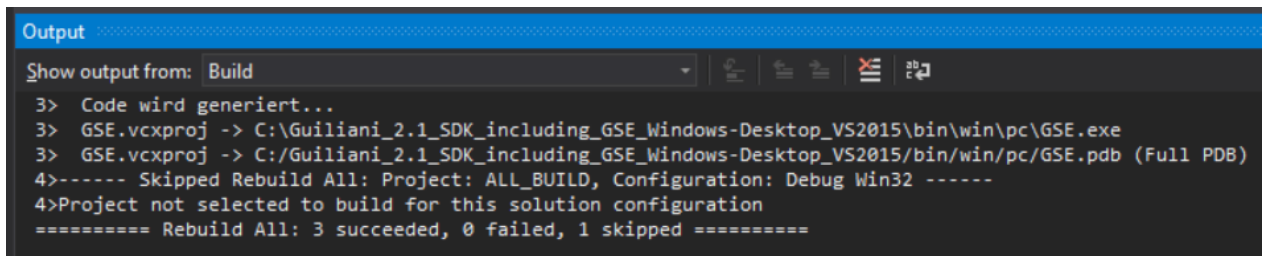


Fig. 11 Rebuild Solution

➔ This should be successful without any errors.



```
Output
Show output from: Build
3> Code wird generiert...
3> GSE.vcxproj -> C:\Guiliani_2.1_SDK_including_GSE_Windows-Desktop_VS2015\bin\win\pc\GSE.exe
3> GSE.vcxproj -> C:\Guiliani_2.1_SDK_including_GSE_Windows-Desktop_VS2015\bin\win\pc\GSE.pdb (Full PDB)
4>----- Skipped Rebuild All: Project: ALL_BUILD, Configuration: Debug Win32 -----
4>Project not selected to build for this solution configuration
===== Rebuild All: 3 succeeded, 0 failed, 1 skipped =====
```

Fig. 12 Successful build



Whenever you add new files to your project (e.g. source- or header-files), you have to rebuild your project. Normally you can use “Build Solution.”

2.2.3. Start the GSE out of Microsoft Visual Studio

▶ Now select “Debug / Start Debugging” from the menu (or press F5) to start the GSE directly out of your development environment (Microsoft Visual Studio).

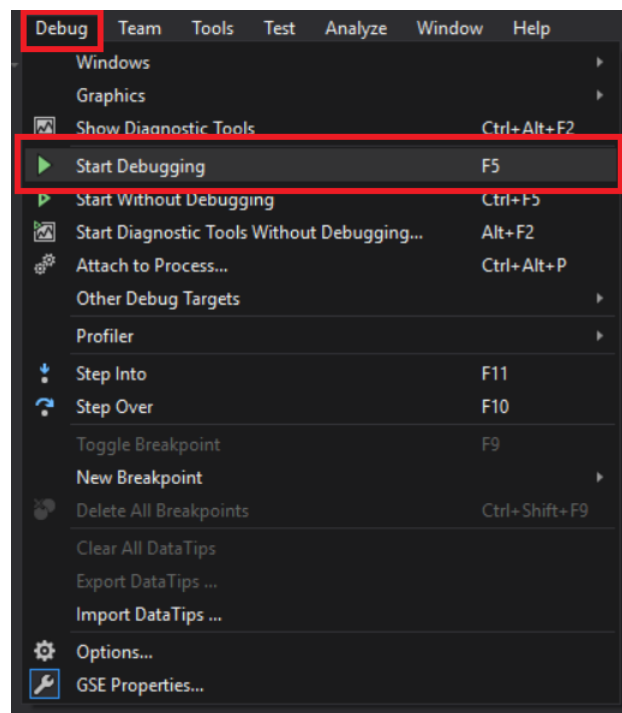


Fig. 13 Start Debugging

2.2.4. Include your GSE project to Microsoft Visual Studio



We will use the “GSE_HowTo” project we created during the “How to 4” lesson. If you don’t have it, you can find the sample solution called “step_by_step” inside the folder called “How to 4 - sample solution”.

- ▶ Copy or move the folder “step_by_step” to the root of your hard disk (C:\).
- ▶ Inside Microsoft Visual Studio right click onto StreamRuntime and select “Set as StartUp Project”. Right click again and select “Properties”.

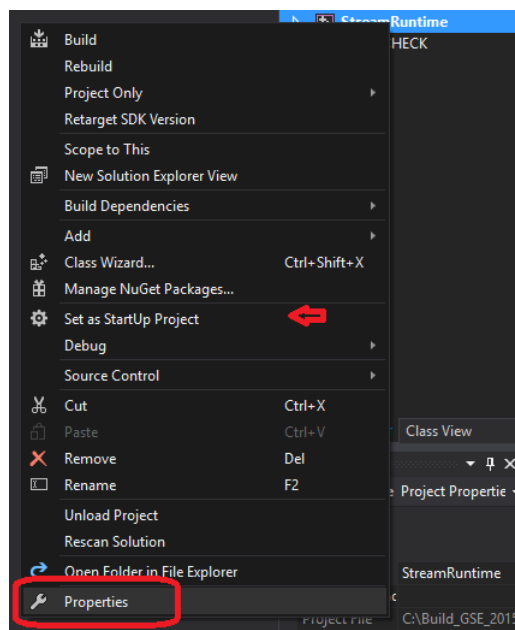


Fig. 14 Set StreamRuntime as StartUp Project and select Properties

Here, check if the Output directory is correct.

- ▶ For this, click onto “General” and look at the path behind “Output Directory”. It has to lead to the folder “resources” inside your Guiliani SDK folder.

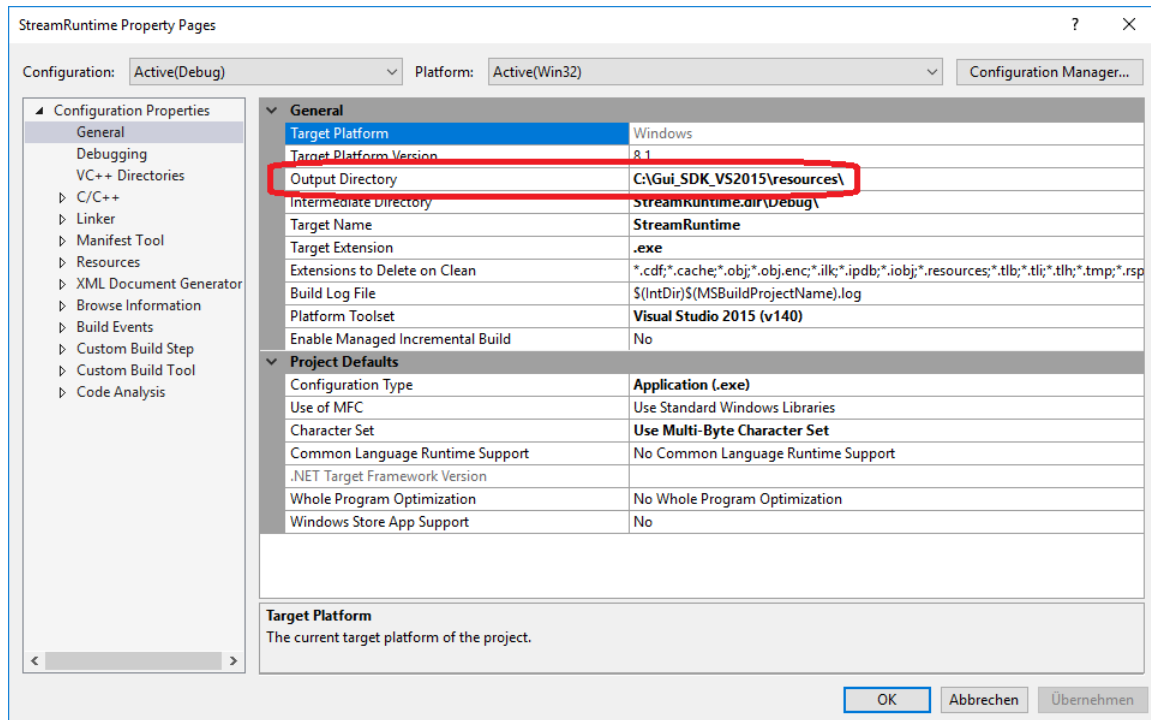


Fig. 15 StreamRuntime Output Directory

Now you have to connect the StreamRuntime to your project.

- ▶ To do this, click onto “Debugging” in the properties page. Click into the field behind “Working Directory” and select “<Browse...>”.

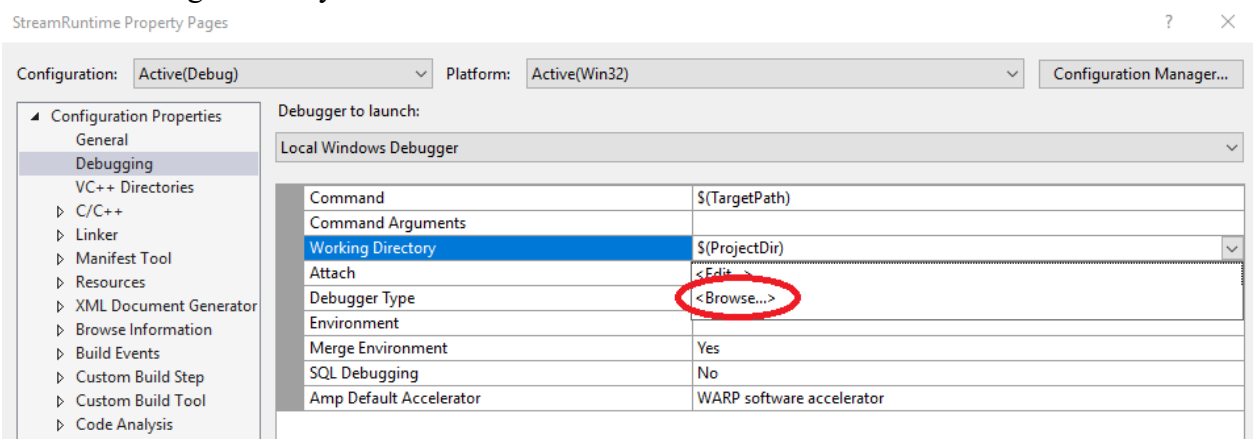


Fig. 16 StreamRuntime Working Directory

- ▶ Navigate to your project (step_by_step) and select the “temp” folder inside your project’s folder. Click “OK”.

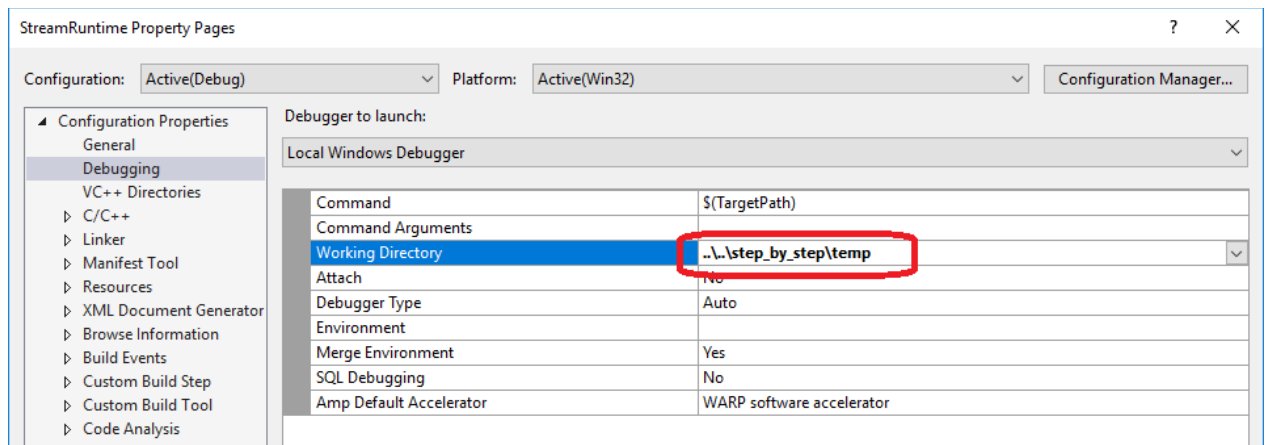


Fig. 17 Set StreamRuntime Working Directory

- ▶ When you now start debugging (menu Debug/Start Debugging or F5) your project will be started.
- ➔ The result should look like this:

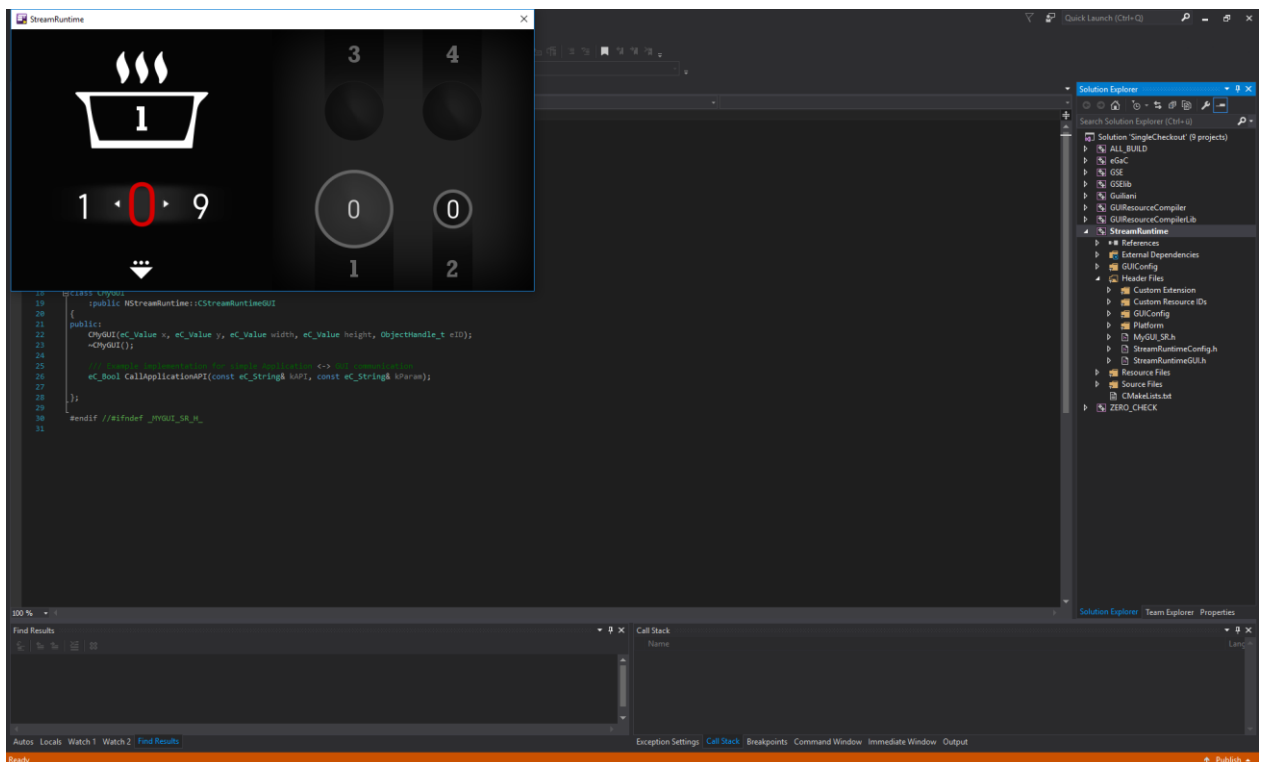


Fig. 18 Run your project

2.3. Step 3: How to continue?

Hopefully we could guide you through the usage of CMake and Microsoft Visual Studio for connecting your step_by_step project to Guiliani SDK GSE.

Please send us (support@guiliani.de) an email and write if you like this step-by-step document or not – and why. Maybe you have some suggestions on how to enhance this guidance or maybe you have found errors – then we would be glad, if you send these to us, too.

Thank you in advance.

And now, go on with our next step_by_step documentation to learn more about using program code to expand your project.

2.3.1. Continuing HowTos

You will find an overview of continuing HowTos in the document “HowTo 0 - an overview of building GSE projects”.

Don't forget to visit our homepage www.guiliani.de to get more information, demos, help, videos and the latest news about guiliani and GSE.

3. Index

A

Assumed knowledge · 5

B

Build directory · 7
Build your solution · 13

C

CMake · 7
 Build directory · 7
Continuing How Tos · 18

D

Documentation conventions · 5

H

How to continue? · 18

I

IDE · 10

P

Prerequisites · 5

S

Short cuts · 6
Solution
 Build · 13