

## Setup build-environment on Windows

Necessary steps to create a build-environment for GSE and Application on a Windows-PC

Product:	Guiliani-SDK
Release version:	2.4
Release date:	April 1, 2021

### Table of contents

1	Introduction .....	2
2	Prerequisites .....	2
3	Using CMake.....	2
3.1	Different Project-Types .....	3
3.1.1	Build GSE and Application in one project.....	3
3.1.2	Build specific Application only.....	3
3.2	Configure and Generate Project .....	5
3.3	Building the project .....	5

# 1 Introduction

This document describes the necessary steps to create a build-environment for the GSE and Application on a Windows-PC

## 2 Prerequisites

The Windows-Version you are using should be the latest version of Windows 10

To setup the build-environment the following programs will be needed:

- CMake 3.13 or higher (“cmake-gui” is highly recommended)
- Microsoft Visual Studio 2015 or higher (Express Editions are supported) or
- MinGW compiler-suite

## 3 Using CMake

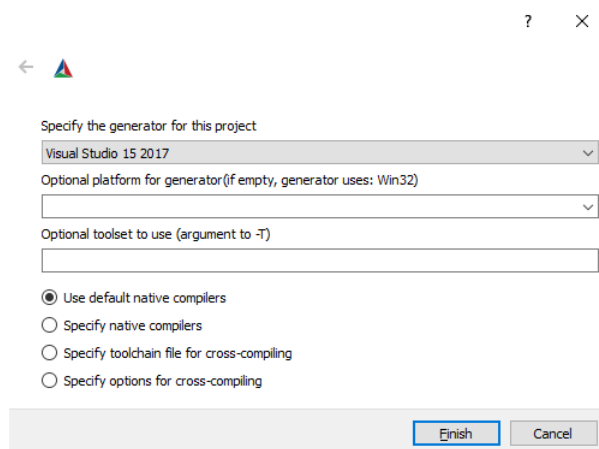
After installing the necessary tools CMake is used for creating a project which can be used inside an IDE to change and compile the GSE and Application.

The procedure in CMake is the following:

- Choose source-location
- Choose build-location where the project will be created
- Click “Configure” and confirm the creation of the build-location if it does not exist yet
- Select the Generator for the project. This can be any of the Generators in the list, but we strongly recommend to choose “Visual Studio XXX” or “MinGW Makefiles”.

**Note: If the selected generator uses 64bit by default, please set it to 32bit.**

- Check and change your configuration to satisfy your needs
- Click “Generate” to create the project



Select the generator for the project

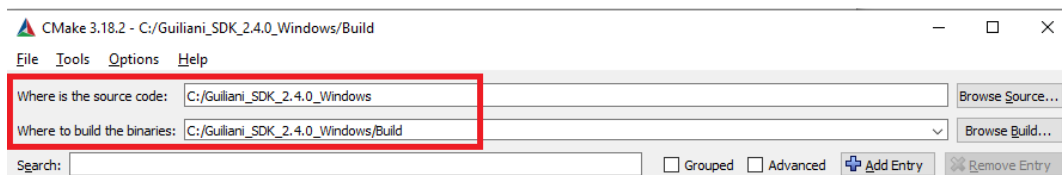
### 3.1 Different Project-Types

For creating a project you have two choices and different configurations depending:

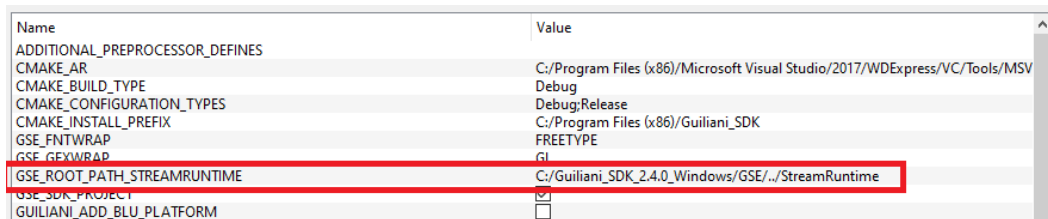
#### 3.1.1 Build GSE and Application in one project

For creating a project which will compile the GSE and Application simultaneously you will choose the root-path of the SDK as the source-location for CMake (e.g. C:\Projects\Guiliani\_SDK\_2.4.0\_Windows).

Additionally you can specify which Application (i.e. which CustomExtensions) should be bound together to the built GSE. By default the Application called StreamRuntime is bound to the GSE.



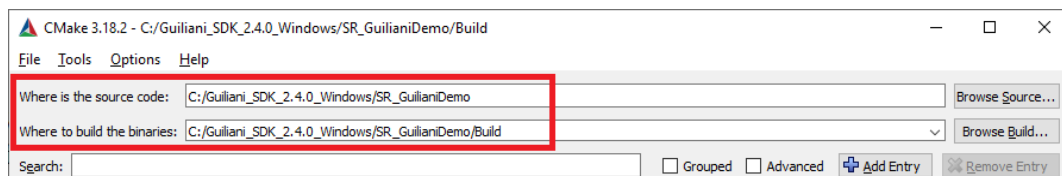
Select source- and destination-location for project



Select Application to be bound to GSE

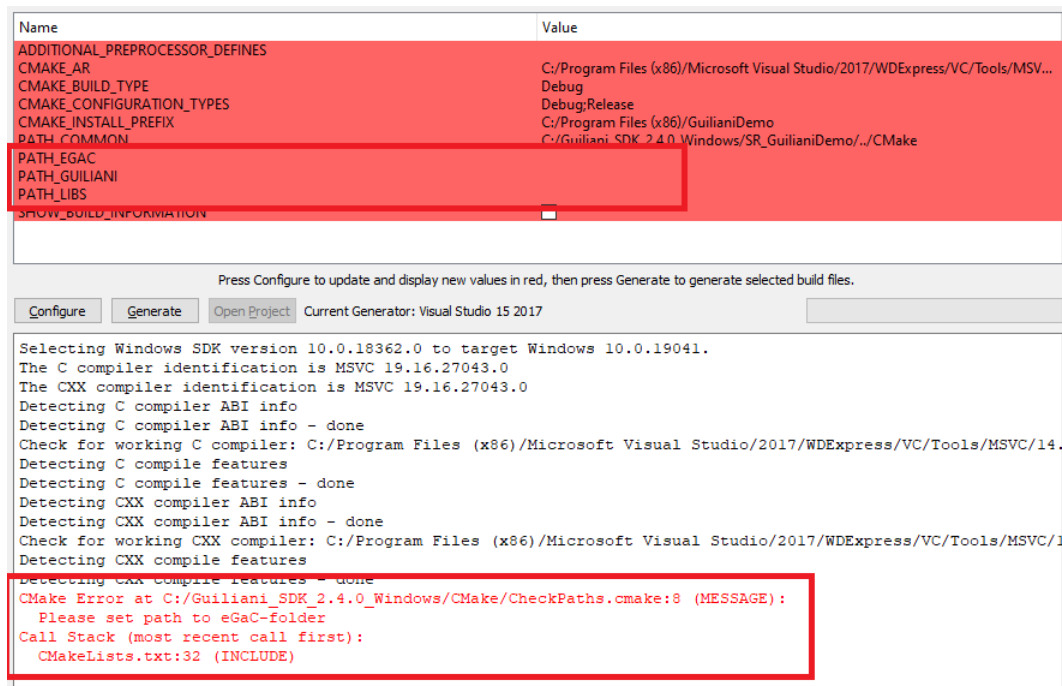
#### 3.1.2 Build specific Application only

When you choose to only build a project for a specific Application you will choose the folder of this Application as the source-location for CMake (e.g. C:\Projects\SR\_GuilianiDemo). The resulting project will not contain the GSE and you will need to do some additional configuration.



Select source- and destination-location for project

After the configuration-process has finished you will see an error-message at the bottom, which says that some folders (e.g. eGaC, Guiliani, etc.) could not be found automatically. You will now have to choose the relevant folders yourself.



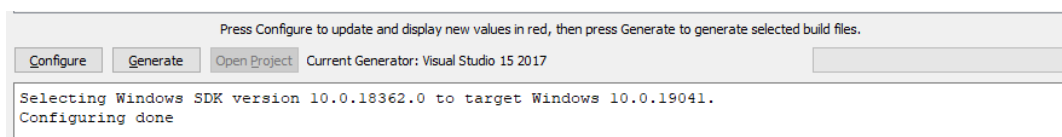
Error-message and missing paths for components necessary for the configuration

Choose the appropriate folders for PATH\_EGAC, PATH\_GUILIANI and PATH\_LIBS and set them to values similar to the ones shown below.



Set the necessary folder-locations manually

After setting the folder-locations by hand click on “Configure” again. Now the message-log at the bottom should not show any errors.



Successful configuration

## 3.2 Configure and Generate Project

After you have the necessary changes in the configuration, click on “Configure” below the list of configuration-values and after that on “Generate”.

**Note: if there are messages in red in the info-box at the bottom, there might be an error during configuration due to invalid values. Please check your configuration in such a case according to the displayed error-message. If you don't know where the error might come from choose to delete the configuration using “File -> Delete Cache” and start over again.**

## 3.3 Building the project

After the project is created you can open it in the appropriate IDE and start working on the source-files and re-compile.