Qt creator/GMlib Windows Guide

Programmes needed:

- CMake
- Git
- Visual Studio
- Qt Creator and Qt libraries

Guide:

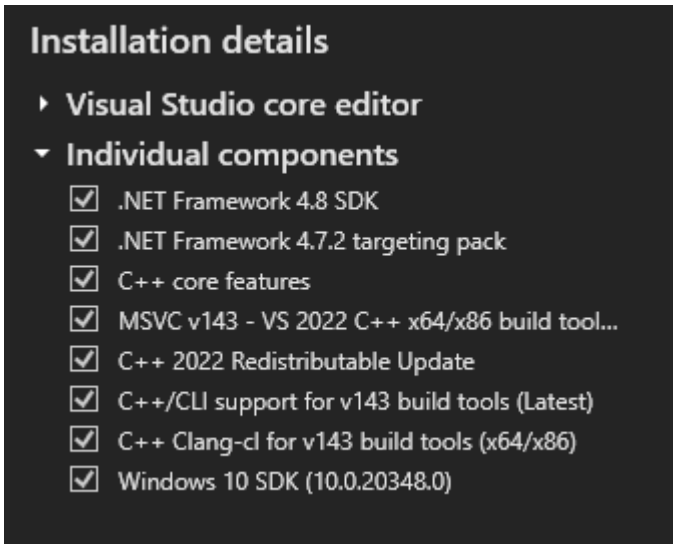
Folder Recommendations:

We recommend clean and simple file- and folder names. For example, space in the directory name may cause CMake to behave odd. Instead of space we recommend using _ . To keep the folder paths short and nice, we recommend having a DEV folder directly in the C:\ drive. Example of folder set up:

- C:\DEV\CourseName\ProjectName
- C:\DEV\vcpkg

Part 1: Download and set up Visual Studio:

- Download and install newest Windows installer for CMake.
- Download and install latest Git source Release for Windows.
- Download Visual Studio Community:
    - Start Visual Studio Installer
    - Do **NOT** select any Workloads
    - Select the “Individual Components” tab
    - Choose the following packages:
        - Under “Compilers, build tools, and runtimes”:
            - MSVC v**XXX** – VS 202**X** C++ x64/x86
            - C++ 202**X** Redistributable Update
            - C++/CLI support for v**XXX** build tools (Latest)
            - C++ Clang-cl for v**XXX** build tools (x64/x86)
        - Under “Development activities”:
            - C++ core features
        - Under “SDKs, libraries and frameworks”:
            - Windows 10 SDK
    - Two .NET components will automatically be chosen due to the choices above.
    - **NOTE**: The v**XXX** in the given packages must be the same, for example all packages use v143.
    - The list chosen packages should look something like this:



- Install the CDB debugger
  - Open “Apps & features” settings in Windows
  - Find the Windows Software Development kit - Windows 10.**XXXX**
  - Select “Modify”
  - Select “Debugging Tools for Windows”
  - Verify that the file ‘C:\Program Files (x86)\Windows Kits\Debuggers\x64\cdb.exe’ is present
- Install the Microsoft C++ Library Manager (vcpkg)
  - Open the Git Bash shell
  - Change directory to where you wish to have the vcpkg tool and fetch vcpkg

    $ cd to_desired_folder

    $ git clone https://github.com/Microsoft/vcpkg.git
- Configure vcpkg
  - Open the Windows command prompt (e.g. Win-R, type cmd.exe)
  - Change directory to the folder where you fetched vcpkg during the installation step

    > cd vcpkg

    > bootstrap-vcpkg.bat

    > vcpkg integrate install
- Install the OpenGL and GLEW library using vcpkg

  > vcpkg install glew:x64-windows

  > vcpkg install opengl:x64-windows
  - **Optional:** Many other libraries, including Google's `gtest` and `benchmark`, can be installed via `vcpkg`. Issue

    > vcpkg list

    to list the currently installed libraries, and

    > vcpkg search

    > vcpkg search

    Will list available packages and filter on a pattern, respectively.

Part 2: Download Qt:

- Apply for Qt Educational license Qt Creator and Qt libraries (using your UiT email)
- Download and execute the Qt installer
  - Click ‘next’ a couple of times in the Wizard to download some packages and select the directory where Qt will be installed

    - Select Components as follows (anything else is not necessary)
        - Under “Qt 5.15.11”:
            - MSVC 2019 64-bit
        - Under “Developer and Designer Tools”:
            - Qt Creator **X.X.X** CDB Debugger Support (Select the latest)
            - Ninja
    - Complete the Wizard and install the selected components
- Verify that there is a default kit “Desktop Qt 5.15.11 MSVC2019 64-bit”
- Select the default kit and verify that the following are configured:
    - Compiler: C++: Microsoft Visual C++ Compiler XX.X (x84-amd64)
    - Debugger: Auto-detected CDB at
    - Qt version: 5.15.11 MSVC2019 64-bit
    - CMake Tool: System CMake at

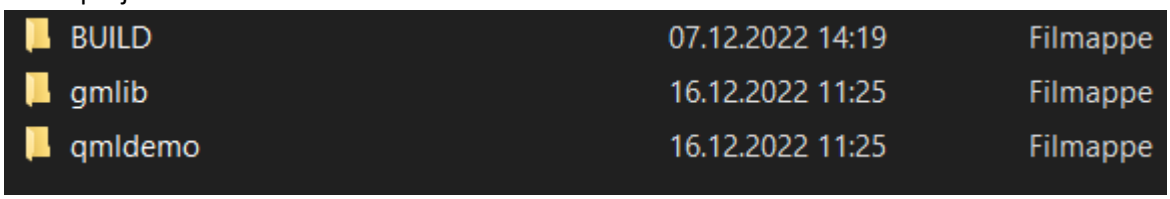
Part 3 Set up Qt and GMlib

- Git clone or download and unzip GMlib and qmldemo to wanted folder

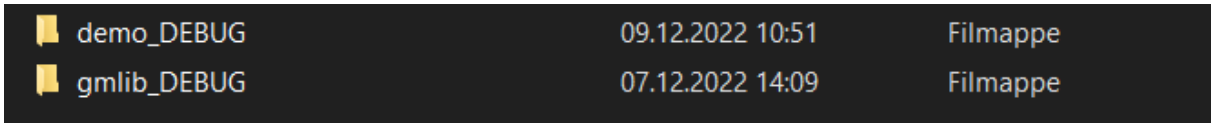
```
$ git clone git@source.coderefinery.org:gmlib/gmlib1/gmlib.git gmlib
```

```
$ git clone git@source.coderefinery.org:gmlib/gmlib1/qmldemo.git demo
```

- Create the corresponding BUILD folder structure
    - In the project folder:



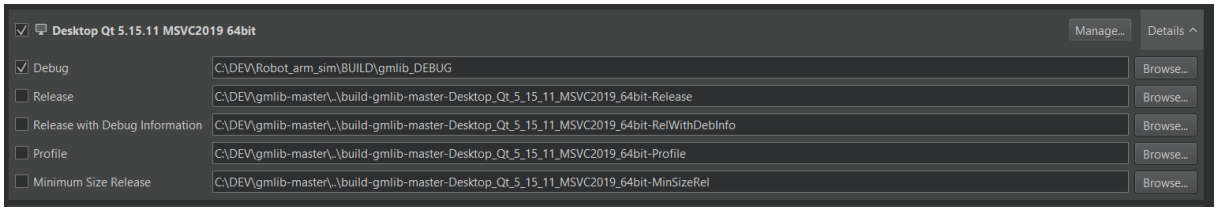
    - In the BUILD folder



    - **NOTE:** The exact name of the BUILD folder and the folders within it is not important. The important part is knowing which the different folders are.

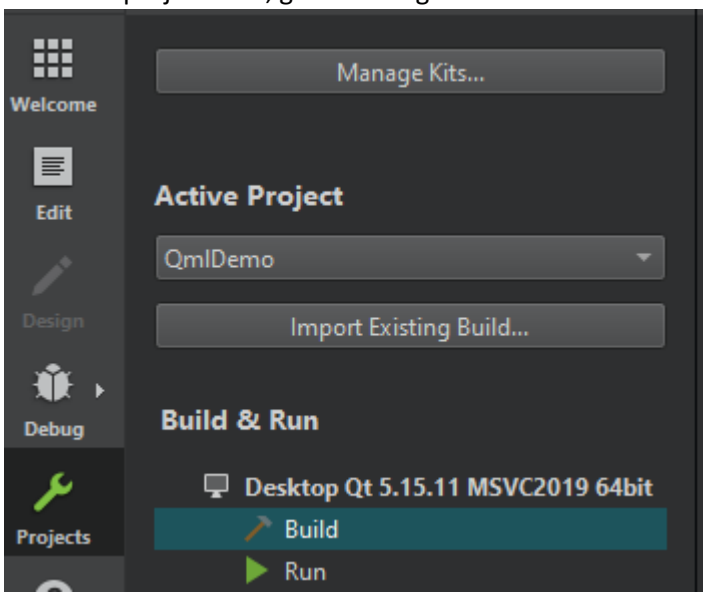
Part 4 Using Qt Creator:

- Create a new session, e.g. named MyProject using the Session Manager and switch to the new session.
- **For gmlib:** From the File menu, choose File -> Open File or Project and choose the CMakeList.txt file inside the gmlib sourse folder (fetch with git)
- De-select all except Debug and Browse to select the corresponding build folder

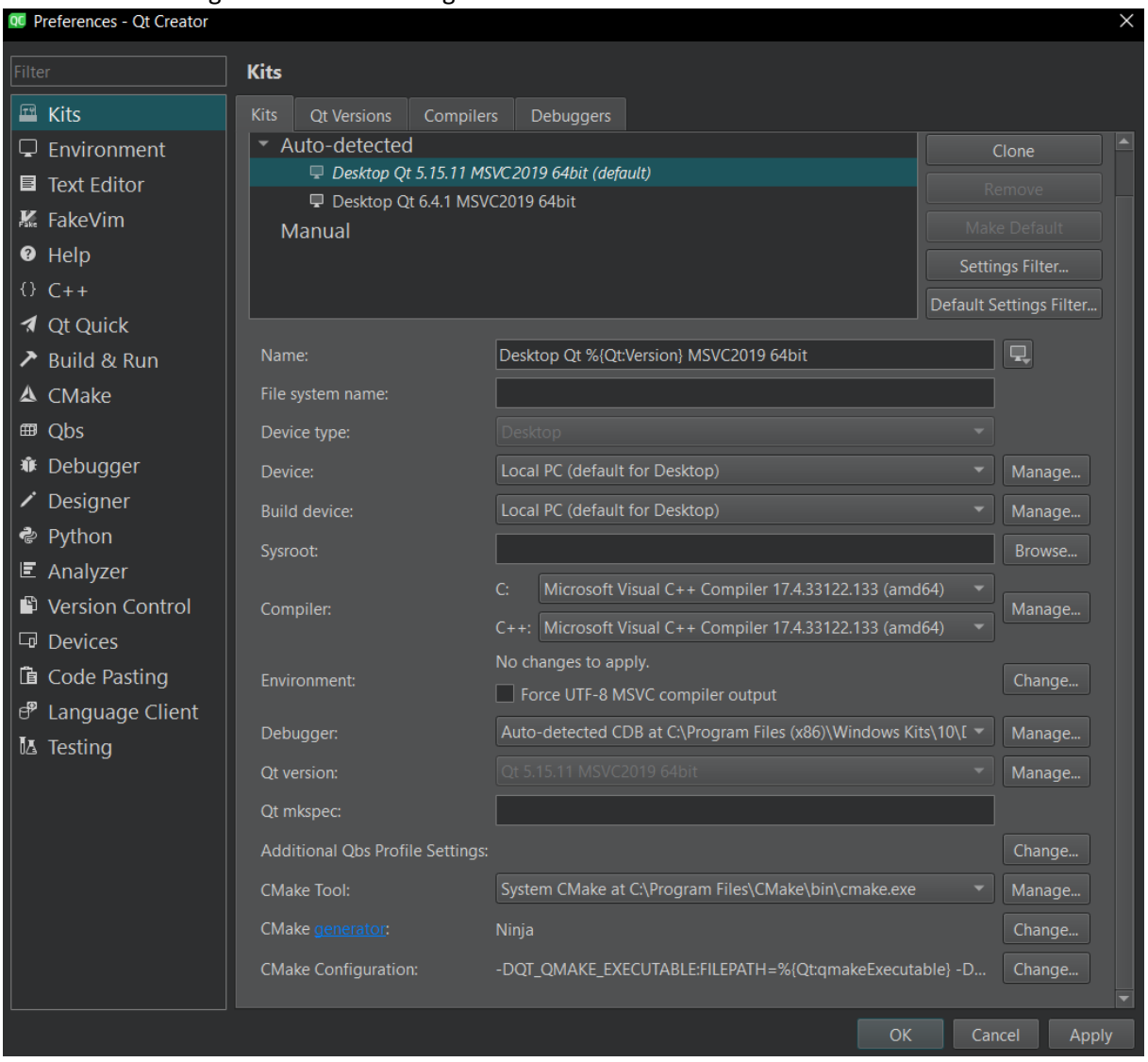


- There will be an error related to vcpkg, this is because we have not yet defined where vcpkg is.

- Go to the projects tab, go to Manage Kits…

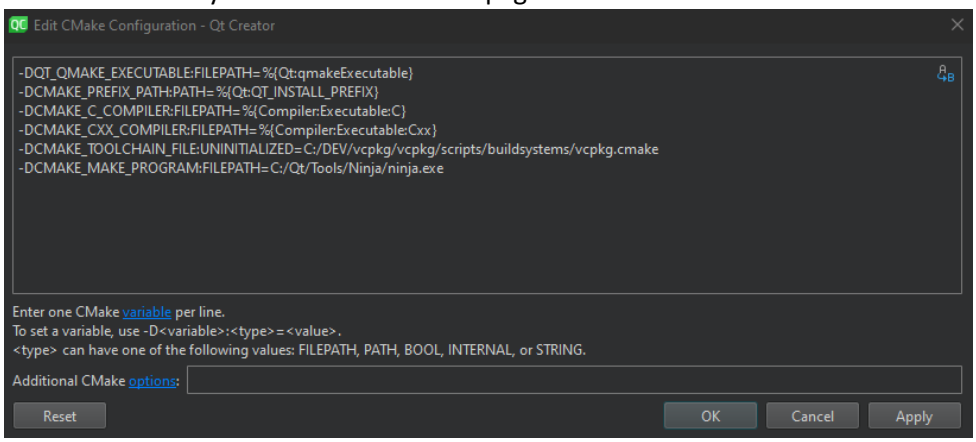


- Click Change… for CMake Configuration.



- Add:
  -DCMAKE_TOOLCHAIN_FILE:UNINITIALIZED=C:/xxx/vcpkg/scripts/buildsystems/vcpkg.cmake

Where xxx is the your location for the vcpkg.



  - Click Apply and OK
- There might be an error related to Ninja
  - Follow the steps for solving the vcpkg error
  - During the Add part, use:
    -DCMAKE_MAKE_PROGRAM:FILEPATH=C:/Qt/Tools/Ninja/ninja.exe
    If Ninja was installed using Qt, it should be located at the given path.