

CMake Merkblatt

Command Line Building

CMake Steps:

Generieren der Build Files:

Command: `cmake [options] -S <path-to-source> -B <path-to-build>`

Bauen des Projects:

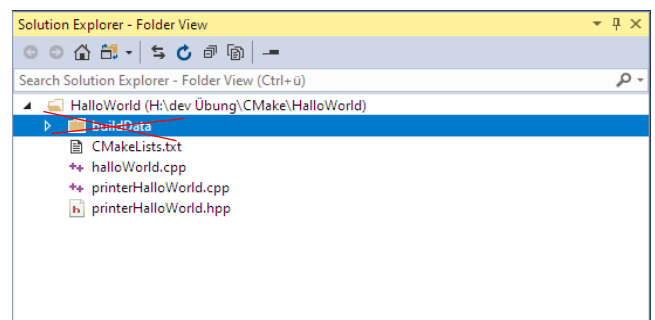
Command: `cmake --build <_path-to-build>`

Ausführen der Exe, in dem Build Protokoll Steht auch immer die Location der Exe

`<_path-to-build>/Debug/<name>.exe`

Anmerkungen:

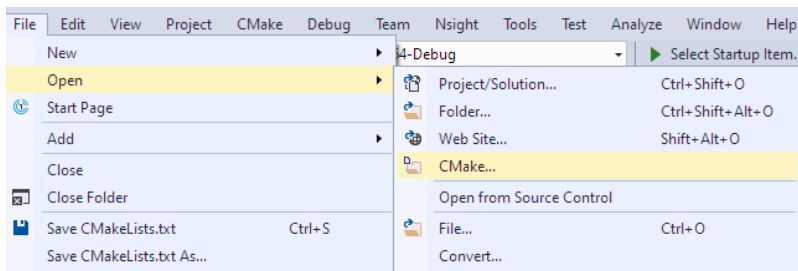
Niemals im Source-Tree bauen, damit Git nicht die Build Daten hochlädt.



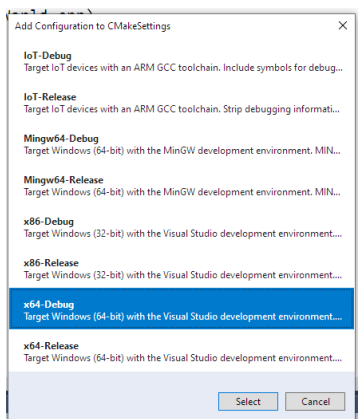
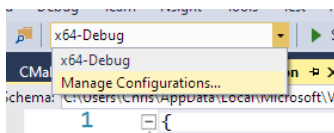
Visual Studio Komponenten



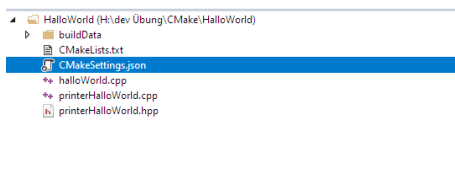
Öffnen eines CMake Projects



Build Settings:



CMakeSettings wird erstellt



Wichtig! CMakeSettings zu Git Ignore hinzufügen

Link für die Einstellungen der CMakeSettings

<https://docs.microsoft.com/de-de/cpp/build/cmake-settings-reference?view=vs-2019>

Meine CMakeSettings.json:

```
{  
  "configurations": [  
    {
```

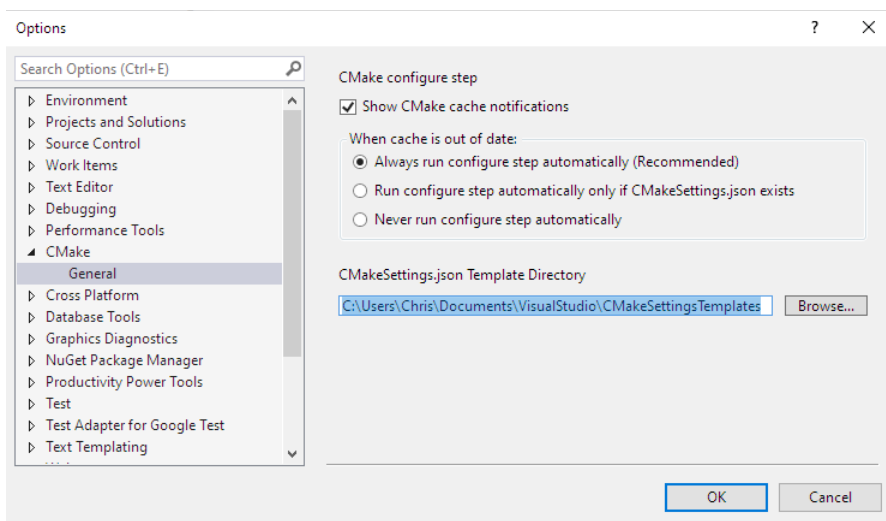
```

"name": "x64-Debug",
"generator": "Ninja",
"configurationType": "Debug",
"inheritEnvironments": [
  "msvc_x64_x64"
],
"buildRoot": "${workspaceRoot}\\..\\build",
"installRoot": "${workspaceRoot}\\..\\installtion",
"cmakeCommandArgs": "",
"buildCommandArgs": "-v",
"ctestCommandArgs": ""
}
]
}

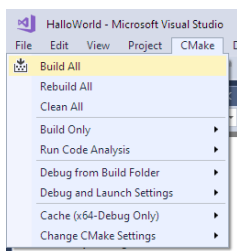
```

Unter Tools->Options->Cmake->Gernal

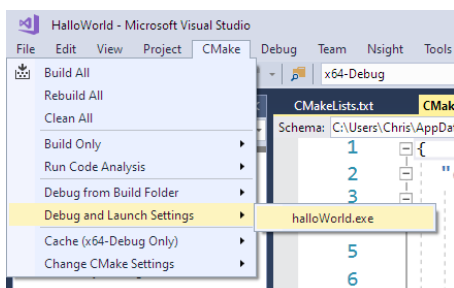
Kann man ein Template für die CMakSettings.json machen



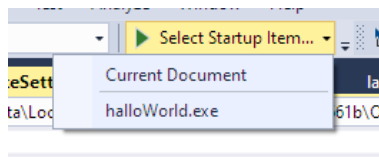
Bauen in Visual Studio:



Falls Was nicht funktioniert bei den CMake Scripten: Clean All



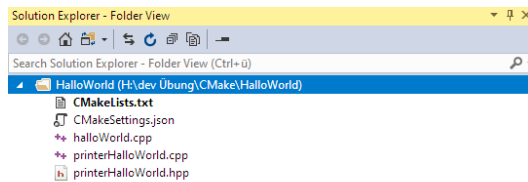
Damit erstellt Visual Studio eine Launch.json damit wie gewohnt Starten kann



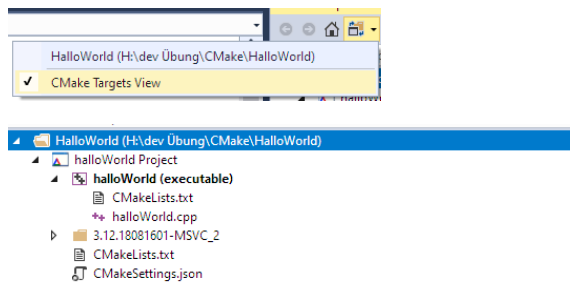
```
configurations": [
```

Ansichten

Ordner Ansicht



Projekt Ansicht



Wenn in VS nichts mehr funktioniert: Benutz das Command Line CMake

Zum CMake Programmieren sind diese Artikel ganz gut (optional)

<https://pabloariasal.github.io/2018/02/19/its-time-to-do-cmake-right/>

<https://rix0r.nl/blog/2015/08/13/cmake-guide/>