

# Installieren von wxWidgets in CodeBlocks

- 1.) **gcc** und wxWidgets-Download-File müssen versionsmäßig zusammenpassen!!!

Für das aktuelle CodeBlocks Version 20.03, das intern als Compiler die MinGW-Portierung des **gcc** in der Version 8.1 (64 bit) verwendet von <https://github.com/wxWidgets/wxWidgets/releases/> folgende zwei Files herunterladen (dabei 3.1.7 durch die aktuelle Version ersetzen):

[https://github.com/wxWidgets/wxWidgets/releases/download/v3.1.7/wxMSW-3.1.7\\_gcc810\\_x64\\_Dev.7z](https://github.com/wxWidgets/wxWidgets/releases/download/v3.1.7/wxMSW-3.1.7_gcc810_x64_Dev.7z)

<https://github.com/wxWidgets/wxWidgets/releases/download/v3.1.7/wxWidgets-3.1.7-headers.7z>

- 2.) Irgendwo ein neues Verzeichnis anlegen (z.B. ganz oben: **D:\wxw-3.1.7** ) und die beiden Zip-Files dort hinein auspacken, sodass das Verzeichnis einen **include**-Ordner und einen **lib**-Ordner (mit Unter-Ordnern) enthält.

- 3.) CodeBlocks starten und ein **neues Konsol-Projekt** (*“Console application”*) mit Sprache C++ anlegen (kein wxWidgets-Projekt, der wxWidgets-Wizard kommt mit der aktuellen wxWidgets-Verzeichnisstruktur nicht zurecht!).

Der Projektname (wird auch Name des **.exe**-Files!) und alle im Projekt verwendeten Dateinamen sollen keine Umlaute oder Zwischenräume enthalten.

Es reicht, die Release-Konfiguration auszuwählen (kein Debug).

- 4.) Zum Testen mein <http://www.computerix.info/examples/wxw-hello.cpp> herunterladen und den Code in das **main.cpp** des Projektes hineinkopieren.

- 5.) In den **Compiler-Einstellungen** des Projektes (links in der Projekt-Ansicht: Rechtsklick auf das Projekt selbst, *“Build options...”*):

- a) Zuerst auf das dritte Karteiblatt *“Search directories”* gehen und dort bei *“Compiler”* mit [Add] zwei Verzeichnisse der wxWidgets-Installation eintragen (in dieser Reihenfolge), und zwar als absoluten Pfad („Keep relative Path“: No):

wxWidgets-Installationspfad\**lib**\gcc810\_x64\_dll\mswu  
wxWidgets-Installationspfad\**include**

- b) Genauso bei *“Search directories”* / *“Linker”* das Lib-Verzeichnis der wxWidgets-Installation eintragen:

wxWidgets-Installationspfad\**lib**\gcc810\_x64\_dll

- c) Dann auf das vorige Karteiblatt *“Linker Settings”* gehen und dort im rechten Feld *“Other linker options:”* Folgendes eintragen:

**-lwxmsw31u\_core -lwxbase31u**

Für komplexe wx-Projekte braucht man eventuell weitere Libraries, aber für unsere einfachen Programme sollten diese beiden ausreichen.

6.) Mit diesen Einstellungen sollte sich eine Musterlösung bzw. dein Programm fehlerfrei compilieren, linken und im CodeBlocks starten lassen.

7.) Wenn man den **.exe** außerhalb von CodeBlocks in einem **cmd**-Fenster bzw. ohne die wxWidgets-Installation starten will (oder z.B. auf einen Stick kopieren will):

wxWidgets liefert leider nur **.dll**'s mit, keine Libraries zum statisch Linken. In „Build options“ / „Compiler Settings“ / „Compiler Flags“ die drei Optionen „Static ...“ ankreuzen hilft daher nicht.

Man muss stattdessen:

- Folgende zwei **.dll** aus `wxWidgets-Installationspfad\lib\gcc810_x64_dll` in dasselbe Verzeichnis kopieren, in dem auch der **.exe** liegt:

**wxbase317u\_gcc810\_x64.dll** und **wxmsw317u\_core\_gcc810\_x64.dll**

- Folgende drei **.dll**'s aus `C:\Programme\CodeBlocks\MinGW\bin` ebenfalls zum **.exe**-File kopieren:

**libgcc\_s\_seh-1.dll** , **libstdc++-6.dll** und **libwinpthread-1.dll**

Falls das nicht reicht und sich Windows bei Starten des Programms immer noch über fehlende **.dll**-Files beschwert:

Die in der Fehlermeldung angezeigten **.dll**'s in den beiden Verzeichnissen oben oder in `...\MinGW\x86_64-w64-mingw32\lib` suchen und ebenfalls in das **.exe**-Verzeichnis kopieren.

Auch mit dem Tool „Dependencies“ (<https://lucasg.github.io/Dependencies/> bzw. <https://github.com/lucasg/Dependencies>) kann man anzeigen, welche **.dll**-Files ein **.exe** wirklich braucht.

## wxWidgets unter Linux und am Mac

- Man braucht das Entwickler-Paket von wxWidgets, meist mit **...devel...** im Namen (nicht nur das Laufzeit-Paket).
- Das Compilieren auf der Befehlszeile sollte wie folgt funktionieren (in einer Zeile!):

```
g++ andere Optionen -o Output-File $(wx-config --cflags)  
alle .cpp-Files $(wx-config --libs)
```