

- `sudo apt install mingw-w64 [g++-mingw-w64] [binutils-mingw-w64]`
- cross compile wxWidgets sources
  - download Latest Stable Release
  - `./configure --prefix=/usr/i686-w64-mingw32 --host=i686-w64-mingw32 (32 bits Windows) --enable-unicode --disable-shared`
  - `make; sudo make install`
- Settings→Compiler→Global compiler settings
  - GNU GCC Compiler→Copy→Xcompiler (or anything you prefer to “Xcompiler”)
  - Search Directories
    - \* Compiler: `/usr/i686-w64-mingw32/include`
    - \* Linker: `/usr/i686-w64-mingw32/lib`
    - \* Ressource compiler: `/usr/i686-w64-mingw32/include`
  - Toolchain executables
    - \* Compiler’s installation directory: `/usr`
    - \* C compiler: `i686-w64-mingw32-gcc`
    - \* C++ compiler: `i686-w64-mingw32-g++`
    - \* Linker for dynamic libs: `i686-w64-mingw32-g++`
    - \* Linker for static libs: `i686-w64-mingw32-ar`
    - \* Debugger: *unsuccessful attempts to (i) install mingw gdb and (ii) call winedb (via Settings→Debugger...)*
    - \* Ressource compiler: ``/usr/i686-w64-mingw32/bin/wx-config --rescomp``
    - \* Make program: `make`
  - settings stored in `~/.config/codeblocks/default.conf`
- Project→Properties
  - →Build targets
    - \* Release (or Debug)→Duplicate→WinRelease (or anything you prefer to “Winrelease”)
    - \* Output filename: `bin/WinRelease/<project>.exe`
    - \* Auto-generate filename extension: off
    - \* Execution working dir: `bin/WinRelease`
    - \* Objects ouput dir: `obj/WinRelease`
  - →Build options
    - \* move (cut/paste) ``wx-config --cflags`` from `<project>`→Compiler settings→Other compiler options to the corresponding boxes in `<project>`-Debug and `<project>`-Release
    - \* idem with ``wx-config --libs`` (`<project>`→Linker settings→Other linker options)
    - \* `<project>`-WinRelease
      - Selected compiler: Xcompiler
      - Compiler settings→Other compiler options: ``/usr/i686-w64-mingw32/bin/wx-config --static --cflags``
      - Linker settings→Other linker options: ``/usr/i686-w64-mingw32/bin/wx-config --static --libs` -static-libgcc -static-libstdc++`
  - settings stored in `<project>.cdb`

**N.B.** Add `-no-pie` to Project→Properties→Build options→`<project>`-Release→Linker settings→Other linker options otherwise Ubuntu thinks it’s a shared library (then `file <project>.exe` says `ELF 64-bit LSB shared object ELF 64-bit LSB executable`).