

Building CGO with Zig

Why Zig which is not C can help with
CGO which is not Go



What is CGO?



“CGO is not Go”

- Dave Cheney (2016)

What is it?

- Interoperability with C or C++
- Allows you to use C / C++ libraries from Go

Why is it (generally aka by Dave Cheney) not recommended?

- Requires full C / C++ tool chain
- Slow C / C++ build
- Cost of crossing programming language and paradigm boundary
- Rely on installation of required libraries or statically link everything to keep single binary
- You now have “two problems”

“But what if I REEEALLY want to?”

- John Hutchison (2025)

GoCV project for Raspberry Pi

- Use OpenCV to detect cat pictures 🐱
- See if they have something in their mouth 🐦 / 🐭
- Block cat door



Sqlite3

- Most used and (still, maybe) performant library mattn/go-sqlite3
- *“This package requires the CGO_ENABLED=1 environment variable if not set by default, and the presence of the gcc compiler.”* - github.com/mattn/go-sqlite (2025)

<https://github.com/mattn/go-sqlite3?tab=readme-ov-file#cross-compile>

Compiling and Cross-Compiling

Compiling to your native OS / CPU architecture (Arch):

- Install C/C++ compiler on your machine, i.e. gcc

```
CGO_ENABLED=1 go build .
```

```
CGO_ENABLED=1 go build -ldflags "-linkmode external -extldflags -static"
```

Cross-Compiling

- Install C/C++ cross-compiler for target OS / Arch
- Maybe, install dependent libraries for target OS / Arch
- If you're a masochist, cross-compile dependent libraries for target OS / Arch

```
CC=arm-linux-gnueabihf-gcc CXX=arm-linux-gnueabihf-g++ \  
CGO_ENABLED=1 GOOS=linux GOARCH=arm GOARM=7 \  
go build .
```

Zig to the rescue?



What even is Zig?

- A systems programming language for writing *blazingly* fast software
 - Safety-wise somewhere between C and Rust
 - No hidden control-flow or memory allocations
- Was influenced by Go and Rust

Stability ... ?



Cross Compiling with Zig



Loris Cro

Posted on Jan 24, 2021 • Edited on May 14, 2022



58



10

Zig Makes Go Cross Compilation Just Work

#zig #go #cpp #systems

Source: <https://dev.to/kristoff/zig-makes-go-cross-compilation-just-work-29ho>

Cross Compiling with Zig

- Zig can be used as a drop-in replacement for Clang/GCC
- No Cross-Compilation tool chains needed

```
CGO_ENABLED=1 GOARCH=amd64 GOOS=windows \  
CC="zig cc -target x86_64-windows" go build
```

References

- <https://ziglang.org/>
- <https://andrewkelley.me/post/zig-cc-powerful-drop-in-replacement-gcc-clang.html>
- <https://jakstys.lt/2022/how-uber-uses-zig/>
- <https://vercel.com/blog/how-we-continued-porting-turborepo-to-rust>

Thanks!

bespinian.io



- What is CGO? => Why CGO is not Go
- Why would you use CGO? => Why did I ask this question?
- What is cross-compiling and how do you traditionally do it?
- What is classically the issue with cross-compiling C compilers?

- What is Zig?
- Why is cross-compiling with Zig the shit?
- Examples:
 - Simple CGO Example with C-Lib => SQLite
 - Simple CGO Example with C-Code
 - GoCV Example with C++-Code and Libs