



Documentation

Tutorials

Guides

Build from source

Embedded

WebAssembly

Using WASI

Using WASM

Additional

Resources

Linux support

macOS support

Tips, Tricks and

Gotchas

Windows support

Contributing

IDE Integration

Debugging

Optimizing binaries

Tinygo flash errors

[Documentation](#) / [Guides](#) / [WebAssembly](#) / Using WASI

Using WASI

How to use TinyGo with the WebAssembly System Interface (WASI).

TinyGo is very useful for compiling programs for use on servers and other edge devices (WASI).

TinyGo programs can run in [Extism](#), [Fastly Compute@Edge](#), [Fermion Spin](#), [tau](#), [wazero](#) and many other WebAssembly runtimes.

Both WASI Preview 1 (wasip1) and WASI Preview 2 (wasip2) are currently supported.

Here is a small TinyGo program for use within a WASI host application:

```
package main

//go:wasmimport yourmodule add
func add(x, y uint32) uint32 {
    return x + y
}

// main is required for the `wasip1` target, even if it isn't used.
func main() {}
```

To compile the above TinyGo program for use on any WASI runtime:

```
GOOS=wasip1 GOARCH=wasm tinygo build -o main.wasm main.go
```

Last modified August 24, 2024: [content/docs: updates for more recent WASM and WASI development info \(60c25b0\)](#)

[View page source](#)

[Edit this page](#)

[Create child page](#)

[Create documentation issue](#)

[Create project issue](#)

