Go 1.23

@peterhellberg

*"We expect **almost all** Go programs to continue to compile and run as before."*

Changes to the language

Go **1.23** makes the “range-over-func” experiment a part of the language.

Changes to the language

The **“range”** clause in a **“for-range”** loop now accepts iterator functions of the following types:

```
func(func() bool)

func(func(K) bool)

func(func(K, V) bool)
```

Tool

Telemetry

Go telemetry is an ***opt-in system***, controlled by the go telemetry command.

GO command

Go command

The new `go env -changed` flag causes the command to print only those settings whose effective value differs from the default value that would be obtained in an empty environment with no prior uses of the `-w` flag.

Go command

The new `go mod tidy -diff` flag causes the command not to modify the files but instead print the necessary changes as a unified diff.

It exits with a **non-zero** code if updates are needed.


Vet

Vet

The `go vet` subcommand now includes the stdversion analyzer, which flags references to symbols that are too new for the version of Go in effect in the referring file.

Trace

Trace

The `trace` tool now better tolerates **partially broken** traces by attempting to recover what trace data it can. This functionality is particularly helpful when viewing a trace that was collected during a program crash, since the trace data leading up to the crash will now be *recoverable* under **most** circumstances.

Runtime

Runtime

The traceback printed by the runtime after an ***unhandled panic*** or ***other fatal error*** now indents the second and subsequent lines of the error message *(for example, the argument to panic)* by a single tab, so that it can be unambiguously distinguished from the stack trace of the first goroutine.

Compiler

Compiler

- The build time overhead to building with **Profile Guided Optimization** has been reduced significantly.
- The compiler in Go 1.23 can now **overlap** the **stack frame slots** of local variables accessed in disjoint regions of a function, which reduces stack usage for Go applications.

Linker

Linker

The linker now disallows using a **`//go:linkname`** directive to refer to internal symbols in the standard library *(including the runtime)* that are not marked with **`//go:linkname`** on their definitions. Similarly, the linker disallows references to such symbols from assembly code.

For backward compatibility, existing usages of **`//go:linkname`** found in a large open-source code corpus remain supported. Any new references to standard library internal symbols will be disallowed.

Standard library

Timer changes

- **`Timers`** and **`Tickers`** that are no longer referred to by the program become eligible for garbage collection immediately, even if their **`Stop`** methods have not been called
- The timer channel associated with a **`Timer`** or **`Ticker`** is now **unbuffered**, with capacity **0**. *The main effect of this change is that Go now guarantees that for any call to a **`Reset`** or **`Stop`** method, no stale values prepared before that call will be sent or received after the call.*

These new behaviors are only enabled when the main Go program is in a module with a go.mod go line using Go **1.23.0** or later.

New unique package

- The new **`unique`** package provides facilities for **canonicalizing** values
  (like “interning” or “hash-consing”).

Iterators

- The new `iter` package provides the basic definitions for working with user-defined iterators.
- The `slices` package adds several functions that work with iterators
- The `maps` package adds several functions that work with iterators

New structs package

- The new **`structs`** package provides types for struct fields that modify properties of the containing struct type such as memory layout.

  In this release, the only such type is **HostLayout** which indicates that a structure with a field of that type has a layout that conforms to host platform expectations.

Minor changes to the library

As usual, there has been **minor** changes to a number of standard library packages, those packages are:

runtime/pprof archive/tar time path/filepath
crypto/x509 runtime/trace database/sql
net/http/httptest debug/elf slices net/http
sync net/netip os syscall net go/ast
encoding/binary crypto/tls sync/atomic
reflect go/types runtime/debug
text/template math/rand/v2 testing/fstest
unicode/utf8

Ports

Ports

- **Darwin**: Requires macOS 11 Big Sur or later; support for previous versions has been discontinued.
- **Linux**: Go 1.24 will require Linux kernel version 3.17 or later, with an exception that systems running 3.10 or later will continue to be supported if the kernel has been patched to support the getrandom system call.
- **OpenBSD**: Experimental support for OpenBSD on 64-bit RISC-V (`GOOS=openbsd`, `GOARCH=riscv64`)
- **ARM64**: New `GOARM64` environment variable, which specifies the minimum target version of the ARM64 architecture at compile time.
- **RISC-V**: New `GORISCV64` environment variable, which selects the RISC-V user-mode application profile for which to compile.
- **WASM**: The `go_wasip1_wasm_exec` script in `GOROOT/misc/wasm` has dropped support for versions of `wasmtime < 14.0.0`.

Learn more

Please read the Go 1.23 Release Notes

https://go.dev/doc/go1.23