

**NAME**

go – tool for managing Go source code

**SYNOPSIS**

**go test** [-c] [-i] [*build flags*] [*packages*] [*flags for test binary*]

**DESCRIPTION**

"Go test" automates testing the packages named by the import paths. It prints a summary of the test results in the format:

```
ok      archive/tar      0.011s
FAIL    archive/zip      0.022s
ok      compress/gzip    0.033s
...
```

followed by detailed output for each failed package.

"Go test" recompiles each package along with any files with names matching the file pattern "\*\_test.go". These additional files can contain test functions, benchmark functions, and example functions. See **go-test-func(7)** for more.

By default, go test needs no arguments. It compiles and tests the package with source in the current directory, including tests, and runs the tests.

The package is built in a temporary directory so it does not interfere with the non-test installation.

**OPTIONS**

In addition to the build flags, the flags handled by 'go test' itself are:

- c      Compile the test binary to pkg.test but do not run it.
- i      Install packages that are dependencies of the test. Do not run the test.

The test binary also accepts flags that control execution of the test; these flags are also accessible by 'go test'. See **go-testflag(7)** for details.

For more about build flags, see **go-build(1)**.

For more about specifying packages, see **go-packages(7)**.

**SEE ALSO**

**go-build(1)**, **go-vet(1)**.

**AUTHOR**

This manual page was written by Michael Stapelberg <stapelberg@debian.org>, for the Debian project (and may be used by others).