

**NAME**

go-install – compile and install packages and dependencies

**SYNOPSIS**

**go install** [*build flags*] [*packages*]

**DESCRIPTION**

Install compiles and installs the packages named by the import paths.

Executables are installed in the directory named by the GOBIN environment variable, which defaults to `$GOPATH/bin` or `$HOME/go/bin` if the GOPATH environment variable is not set. Executables in \$GOROOT are installed in `$GOROOT/bin` or `$GOTOOLDIR` instead of `$GOBIN`. Cross compiled binaries are installed in `$GOOS_$GOARCH` subdirectories of the above.

If the arguments have version suffixes (like `@latest` or `@v1.0.0`), “go install” builds packages in module-aware mode, ignoring the `go.mod` file in the current directory or any parent directory, if there is one. This is useful for installing executables without affecting the dependencies of the main module. To eliminate ambiguity about which module versions are used in the build, the arguments must satisfy the following constraints:

- Arguments must be package paths or package patterns (with “...” wildcards). They must not be standard packages (like `fmt`), meta-patterns (`std`, `cmd`, `all`), or relative or absolute file paths.
- All arguments must have the same version suffix. Different queries are not allowed, even if they refer to the same version.
- All arguments must refer to packages in the same module at the same version.
- Package path arguments must refer to main packages. Pattern arguments will only match main packages.
- No module is considered the “main” module. If the module containing packages named on the command line has a `go.mod` file, it must not contain directives (`replace` and `exclude`) that would cause it to be interpreted differently than if it were the main module. The module must not require a higher version of itself.
- Vendor directories are not used in any module. (Vendor directories are not included in the module zip files downloaded by ‘go install’.)

If the arguments don’t have version suffixes, “go install” may run in module-aware mode or GOPATH mode, depending on the `GO111MODULE` environment variable and the presence of a `go.mod` file. See ‘go help modules’ for details. If module-aware mode is enabled, “go install” runs in the context of the main module.

When module-aware mode is disabled, non-main packages are installed in the directory `$GOPATH/pkg/$GOOS_$GOARCH`. When module-aware mode is enabled, non-main packages are built and cached but not installed.

Before Go 1.20, the standard library was installed to `$GOROOT/pkg/$GOOS_$GOARCH`. Starting in Go 1.20, the standard library is built and cached but not installed. Setting `GODEBUG=installgoroot=all` restores the use of `$GOROOT/pkg/$GOOS_$GOARCH`.

For more about build flags, see ‘go help build’.

For more about specifying packages, see ‘go help packages’.

**SEE ALSO**

**go build**(1), **go get**(1), **go clean**(1).

**AUTHOR**

This manual page was created using `help2man` and afterwards updating the output. It is maintained by the Debian Go Compiler Team <[team+go-compiler@tracker.debian.org](mailto:team+go-compiler@tracker.debian.org)> for the Debian project (and may be used by others).