



Full credit is given to the above companies including the Operating System (OS) that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'go-packages.7'

\$ man go-packages.7

GO-PACKAGES(7) Miscellaneous Information Manual GO-PACKAGES(7)

NAME

go - tool for managing Go source code

DESCRIPTION

Many commands apply to a set of packages:

go action [packages]

Usually, [packages] is a list of import paths.

An import path that is a rooted path or that begins with a . or .. element is interpreted as a file system path and denotes the package in that directory.

Otherwise, the import path P denotes the package found in the directory DIR/src/P for some DIR listed in the GOPATH environment variable (For more details see: 'go help gopath').

If no import paths are given, the action applies to the package in the current directory.

There are four reserved names for paths that should not be used for packages to be built with the go tool:

? "main" denotes the top-level package in a stand-alone executable.

? "all" expands to all packages found in all the GOPATH trees. For example, 'go list all' lists all the packages on the local system. When using modules, "all" expands to all packages in the main module and their dependencies, including dependencies needed by tests of any of those.

? "std" is like all but expands to just the packages in the standard Go library.

? "cmd" expands to the Go repository's commands and their internal libraries.

Import paths beginning with "cmd/" only match source code in the Go repository.

An import path is a pattern if it includes one or more "..." wildcards, each of which can

match any string, including the empty string and strings containing slashes. Such a pattern expands to all package directories found in the GOPATH trees with names matching the patterns.

To make common patterns more convenient, there are two special cases. First, `/...` at the end of the pattern can match an empty string, so that `net/...` matches both `net` and packages in its subdirectories, like `net/http`. Second, any slash-separated pattern element containing a wildcard never participates in a match of the "vendor" element in the path of a vendored package, so that `./...` does not match packages in subdirectories of `./vendor` or `./mycode/vendor`, but `./vendor/...` and `./mycode/vendor/...` do. Note, however, that a directory named `vendor` that itself contains code is not a vendored package: `cmd/vendor` would be a command named `vendor`, and the pattern `cmd/...` matches it. See golang.org/s/go15vendordor for more about vendoring.

An import path can also name a package to be downloaded from a remote repository. Run `'go help importpath'` for details.

Every package in a program must have a unique import path. By convention, this is arranged by starting each path with a unique prefix that belongs to you. For example, paths used internally at Google all begin with `'google'`, and paths denoting remote repositories begin with the path to the code, such as `'github.com/user/repo'`.

Packages in a program need not have unique package names, but there are two reserved package names with special meaning. The name `main` indicates a command, not a library. Commands are built into binaries and cannot be imported. The name `documentation` indicates documentation for a non-Go program in the directory. Files in package documentation are ignored by the `go` command.

As a special case, if the package list is a list of `.go` files from a single directory, the command is applied to a single synthesized package made up of exactly those files, ignoring any build constraints in those files and ignoring any other files in the directory.

Directory and file names that begin with `."` or `"_"` are ignored by the `go` tool, as are directories named `"testdata"`.

AUTHOR

This manual page was written by Michael Stapelberg <stapelberg@debian.org> and is maintained by the Debian Go Compiler Team <team+go-compiler@tracker.debian.org> based on the output of `'go help packages'` for the Debian project (and may be used by others).