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 (see 'go help gopath').

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

The special import path "all" expands to all package directories found in all the GOPATH trees. For example, 'go list all' lists all the packages on the local system.

The special import path "std" is like all but expands to just the packages in the standard Go library.

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. As a special case, x/... matches x as well as x's subdirectories. For example, net/... expands to net and packages in its subdirectories.

An import path can also name a package to be downloaded from a remote repository. Run 'go help remote' 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 'code.google.com/p/project'.

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.

AUTHOR

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