go-get-install-deprecation

- Overview
- What to use instead
- Why this is happening

Overview

Starting in Go 1.17, installing executables with `go get` is deprecated. `go install` may be used instead.

In Go 1.18, `go get` will no longer build packages; it will only be used to add, update, or remove dependencies in `go.mod`. Specifically, `go get` will always act as if the `-d` flag were enabled.

What to use instead

To install an executable in the context of the current module, use `go install`, without a version suffix, as below. This applies version requirements and other directives from the `go.mod` file in the current directory or a parent directory.

```
go install example.com/cmd
```

To install an executable while ignoring the current module, use `go install` *with* a version suffix like `@v1.2.3` or `@latest`, as below. When used with a version suffix, `go install` does not read or update the `go.mod` file in the current directory or a parent directory.

```
# Install a specific version.
go install example.com/cmd@v1.2.3

# Install the highest available version.
go install example.com/cmd@latest
```

In order to avoid ambiguity, when `go install` is used with a version suffix, all arguments must refer to `main` packages in the same module at the same version. If that module has a `go.mod` file, it must not contain directives like `replace` or `exclude` that would cause it to be interpreted differently if it were the main module. The module's `vendor` directory is not used.

See `go install` for details.

Why this is happening

Since modules were introduced, the `go get` command has been used both to update dependencies in `go.mod` and to install commands. This combination is frequently confusing and inconvenient: in most cases, developers want to update a dependency or install a command but not both at the same time.

Since Go 1.16, `go install` can install a command at a version specified on the command line while ignoring the `go.mod` file in the current directory (if one exists). `go install` should now be used to install commands in most cases.

`go get`'s ability to build and install commands is now deprecated, since that functionality is redundant with `go install`. Removing this functionality will make `go get` faster, since it won't compile or link packages by default. `go get` also won't report an error when updating a package that can't be built for the current platform.

See proposal #40276 for the full discussion.

□□□ #2
□ □□□ □□□ 2026-10-27 04:26:29 UTC
□ □□□ □□□ 2026-10-27 04:26:29 UTC