

[🏠 首页 \(/\)](#) / [网友博文 \(/articles\)](#) / [Go项目目录管理](#)

Go项目目录管理

📅 2015-01-17 16:15 👤 sincerely

👁 阅读 8150 次 ❤️ 1 人喜欢 💬 1 条评论 (/articles/2233#commentForm) ☆ 收藏

在 Go 的官网文档How to Write Go Code (<https://golang.org/doc/code.html#Workspaces>)中，已经介绍了 Go 的项目目录一般包含以下几个：

- **src** 包含项目的源代码文件；
- **pkg** 包含编译后生成的包/库文件；
- **bin** 包含编译后生成的可执行文件。

可以通过下面的例子来说明工程目录的组织管理。(Windows 7 64位, go version go1.3.3 windows/amd64)

1. 创建一个库文件

创建一个库文件 **a.go** 并保存在**scr**目录的一个子目录下面。

```
package myfunc

import "fmt"

func Afunc(str string) {
    fmt.Println("a.go is package mufunc.")
    fmt.Println(str)
}
```

这时候目录结构如下：

```
<dirtest>
|--<src>
    |--<mufunc>
        |--a.go
```

2. 创建main

关于**main**包的位置，可以参照参考资料2，个人建议放在 **scr/main** 下面，毕竟官方推荐包名和文件存放的文件夹名称最好相同(虽然包名和文件夹名可以不相同，也就是说一个文件夹下可以包含多个包的 **.go** 文件)。

```
package main

import "myfunc"

func main() {
    myfunc.Afunc("b.go is package main.")
}
```

这时候目录结构如下：

```
<dirtest>
|--<src>
    |--<mufunc>
        |--a.go
    |--<main>
        |--b.go
```

3. 使用go build

如果这时候使用 **go build**，你会发现下面的输出：

```
E:\dirtest>go build src\main\b.go

src\main\b.go:3:8: cannot find package "myfunc" in any of:
    C:\Program Files\go\src\pkg\myfunc (from $GOROOT)
    D:\GoLang\src\myfunc (from $GOPATH)
```

从输出中我们可以看到，Go 先是从 **\$GOROOT** 中查找包 **myfunc**，如果没找到就从 **\$GOPATH** 中查找，结果都没有找到，我们可以使用 **go env** 输出 Go 的环境变量设置：

```
E:\dirtest>go env

set GOARCH=amd64
set GOBIN=
set GOCHAR=6
set GOEXE=.exe
set GOHOSTARCH=amd64
set GOHOSTOS=windows
set GOOS=windows
set GOPATH=D:\GoLang
set GORACE=
set GOROOT=C:\Program Files\go
set GOTOOOLDIR=C:\Program Files\go\pkg\tool\windows_amd64
set CC=gcc
set GOGCCFLAGS=-m64 -mthreads -fmessage-length=0
set CXX=g++
set CGO_ENABLED=1
```

显然 E:\dirtest 这个目录没有加到 \$GOPATH 中，在环境变量中添加该目录：



保存后，重新执行(可能需要重新打开控制台，让环境变量生效) `go build`，就在当前目录生成了一个可执行文件 `b.exe`。

```
E:\dirtest\src\main>go env
set GOARCH=amd64
set GOBIN=
set GOCHAR=6
set GOEXE=.exe
set GOHOSTARCH=amd64
set GOHOSTOS=windows
set GOOS=windows
set GOPATH=D:\GoLang;E:\dirtest
set GORACE=
set GOROOT=C:\Program Files\go
set GOTOOOLDIR=C:\Program Files\go\pkg\tool\windows_amd64
set CC=gcc
set GOGCCFLAGS=-m64 -mthreads -fmessage-length=0
set CXX=g++
set CGO_ENABLED=1

E:\dirtest>go build src\main\b.go
E:\dirtest>dir

E:\dirtest 的目录

2015/01/13  23:11    <DIR>        .
2015/01/13  23:11    <DIR>        ..
2015/01/13  23:11    1,958,912    b.exe
2015/01/13  22:52    <DIR>        src

E:\dirtest>b.exe
a.go is package mufunc.
b.go is package main.
```

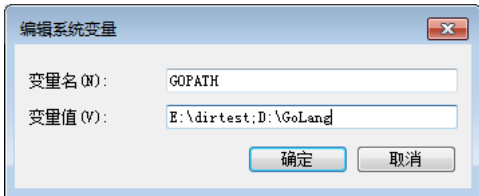
虽然成功运行，但是没有按照期待的那样生成在 *bin* 目录下面。为了达到这样的效果，你需要 `go install`。注意 `go install` 是针对 `package`，而不是针对单个 `.go` 文件。

但是如果是当前状态执行 `go install`，虽然可以成功，但你会发现，并没有在项目根目录 E:\dirtest 中创建 `bin/main.exe`，反而是在 D:\GoLang 中创建了。

如果对 main 包执行 `go install` 呢？

```
E:\dirtest\src\main>go install
go install: no install location for E:\dirtest\src\main: hidden by D:\GoLang\src\main
```

可以看到，输出提示当前目录被隐藏。显然这个顺序是对应 \$GOPATH 的设置，把 \$GOPATH 中的路径顺序改一下：



然后在执行 `go install myfunc`，发现成功地在`pkg`目录下面生成了 `myfunc.a`。同样执行 `go install main`，也成功的在`bin`目录下生成了 `main.exe`。此时的目录结构如下：

```
<dirtest>
|--<src>
    |--<mufunc>
        |--a.go
    |--<main>
        |--b.go
|--<pkg>
    |--<windows_amd64>
        |--myfunc.a
|--<bin>
    |--main.exe
```

现在就算是成功完成了一个示例“项目”吧...

4. 常见错误

除了上面的步骤中出现的错误，其实工程目录管理稍有不慎，就会出现其他问题，例如：

1. 一个文件夹下面包含多个不同包的源文件。也就是把 `a.go` 和 `b.go` 都放到 `myfunc` 目录下面会是什么情况呢？

这时候的目录结构如下：

```
<dirtest>
|--<src>
    |--<mufunc>
        |--a.go
        |--b.go
```

那么执行 `go install` 和 `go build`，甚至 `go run` 都会是相同的错误：

```
E:\dirtest\src\myfunc>go install
can't load package: package myfunc: found packages myfunc (a.go) and main (b.go)
in E:\dirtest\src\myfunc

E:\dirtest\src\myfunc>go build
can't load package: package myfunc: found packages myfunc (a.go) and main (b.go)
in E:\dirtest\src\myfunc

E:\dirtest\src\myfunc>go run b.go
b.go:3:8: found packages myfunc (a.go) and main (b.go) in E:\dirtest\src\myfunc
```

从参考资料3中可以看到，每个子目录中只能存在一个 **package**，否则编译时会报错，所以一个子目录下面不能包含多个不同包的源文件。

2. 一个项目能包含多个 `main()` 吗？

简单测试下，创建一个 `c.go`，并使用 `myfunc` 包(没有导入其他包的情况类似)：

```
package main
import "fmt"
import "myfunc"

func main() {
    fmt.Println("This is single c.go")
    myfunc.Afunc("c.go is also package main.")
}
```

执行相应的命令，结果如下：

```
E:\dirtest\src\main>go build
# main
.\c.go:4: main redeclared in this block
      previous declaration at .\b.go:5

E:\dirtest\src\main>go build c.go
# 成功, 当前目录下生成了c.exe

E:\dirtest\src\main>go install
# main
.\c.go:4: main redeclared in this block
      previous declaration at .\b.go:5

E:\dirtest\src\main>go install c.go
go install: no install location for .go files listed on command line (GOBIN not set)

E:\dirtest\src\main>go run c.go
This is single c.go
a.go is package mufunc.
c.go is also package main.
```

显然只能是 `go run` 和 `go build c.go` 可行。如果把 `c.go` 移到单独的目录下面呢：

```
E:\dirtest\src\cmain>dir

E:\dirtest\src\cmain 的目录

2015/01/14  11:27    <DIR>        .
2015/01/14  11:27    <DIR>        ..
2015/01/14  11:24                147 c.go

E:\dirtest\src\cmain>go build

E:\dirtest\src\cmain>go install
```

均可以执行成功。 `go install` 在 `bin` 目录下面生成了对应的 `exe` 文件。看来还是目录管理的问题。

3. go install: no install location for .go files listed on command line (GOBIN not set)

从上面的示例输出中就能看到，使用 `go install` 针对单个文件时，就会出现这个错误。默认情况下如果设置了 `$GOROOT` 和 `$GOPATH`，就会依次寻找 `$GOROOT/bin` 和 `$GOPATH/bin`。那么我们如果自定义设置了 `$GOBIN=E:\dirtest\bin` 之后会怎样？



```
E:\dirtest\src\cmain>go env
set GOARCH=amd64
set GOBIN=E:\dirtest\bin
set GOCHAR=6
set GOEXE=.exe
set GOHOSTARCH=amd64
set GOHOSTOS=windows
set GOOS=windows
set GOPATH=E:\dirtest;D:\GoLang
set GORACE=
set GOROOT=C:\Program Files\go
set GOTOLDR=C:\Program Files\go\pkg\tool\windows_amd64
set CC=gcc
set GOGCCFLAGS=-m64 -mthreads -fmessage-length=0
set CXX=g++
set CGO_ENABLED=1

E:\dirtest\src\cmain>go install c.go
# 成功在 E:\dirtest\bin 下面生成 c.exe
```

虽然成功了，但是 `go install` 应该是作用于包级别，而非单个文件。

4. go build、go install 和 go run 的区别

详细的可以查看参考资料4，这里简单说一下：

- `go build` 编译包，如果是 `main` 包则在当前目录生成可执行文件，其他包不会生成 `.a` 文件；

- go install 编译包，同时复制结果到 \$GOPATH/bin，\$GOPATH/pkg 等对应目录下；
- go run gofiles... 编译列出的文件，并生成可执行文件然后执行。注意只能用于 main 包，否则会出现 go run: cannot run non-main package 的错误。

此外，go run 是不需要设置 \$GOPATH 的，但 go build 和 go install 必须设置。go run 常用来测试一些功能，这些代码一般不包含在最终的项目中。

5. 总结

1. 一定要管理好目录
2. 多个项目最好都在一个 \$GOPATH 下面，即 src/proj1，src/proj2，etc
3. 尽量使用 go install，这样能够规范项目整体结构

6. 参考资料

1. Golang项目目录结构组织 (<http://www.cnblogs.com/wlts/archive/2013/06/01/3112004.html>)
2. 关于main包放在哪的问题 (<http://blog.studygolang.com/2012/12/%E5%85%B3%E4%BA%8Emain%E5%8C%85%E6%94%BE%E5%9C%A8%E5%93%AA%E7%9A%84%E9%>)
3. 关于golang中包（package）的二三事儿 (<http://www.cnblogs.com/dajianshi/p/3596492.html>)
4. Running multi-file main package ([https://groups.google.com/forum/?fromgroups=#!searchin/golang-nuts/why\\$20use\\$20go\\$20install/golang-nuts/qysy2bM_o1l/khIRe3AJ_wEJ](https://groups.google.com/forum/?fromgroups=#!searchin/golang-nuts/why$20use$20go$20install/golang-nuts/qysy2bM_o1l/khIRe3AJ_wEJ))

本文来自：博客园 ([/wr?u=http://www.cnblogs.com](/?u=http://www.cnblogs.com))
感谢作者：sincerely
查看原文：Go项目目录管理 ([/wr?u=http%3a%2f%2fwww.cnblogs.com%2fsincerely%2fp%2f4230572.html](/?u=http%3a%2f%2fwww.cnblogs.com%2fsincerely%2fp%2f4230572.html))

♥ 1人喜欢

★ 收藏

(<http://www.jiathis.com/share?uid=1895190>) 1

猜你喜欢

换一换



golang安装和部署



go语言工程管理



LiteIDE下设置GOPATH



Go/Golang 语言学习
笔记总结



Golang学习笔记：包
制作



Eclipse 安装Go插件
(Ubuntu)

golang在ubuntu-12.04和centos-6.5上的安装

go 编译问题 | Golang中文社区(Go语言构建) | Go语言中... 学习园地

Golang 工程自管理

Golang项目目录结构组织

GO环境设置

Go语言，编译时不生成bin目录的原因

[go语言]环境配置

golang小程序试验（四）

Go项目的目录结构

go 命令行


用gvm管理Go项目的workspace

go install 的工作方式

百度推荐

« (</articles/2232>) 上一篇: [Go错误记录 \(/articles/2232\)](/articles/2232)
» (</articles/2234>) 下一篇: [Google IP 最新地址 \(/articles/2234\)](/articles/2234)


1 条评论



lsideyu (/user/lsideyu):

我不得不说大部分这篇文档你写的很好，但是有一个错误，至少在我这里是错误的，我要指出来，就是mufunc的文件夹名称要改成myfunc 不然不能通过go build命令的 这样新手才能比较完美的搭建好环境，再次感谢您的努力和工鞋。

(/user/lsideyu)

1楼, 2015-07-01  回复

文章点评：（您需要 登录 后才能评论 没有账号 (</user/register>) ？）

[编辑](#) [预览](#)

我有话要说.....

- 支持 Markdown 格式, **粗体**、~~删除线~~、`单行代码`
- 支持 @ 本站用户；支持表情（输入：提示），见 [Emoji cheat sheet \(http://www.emoji-cheat-sheet.com/\)](http://www.emoji-cheat-sheet.com/)

提交

- Go最新资料汇总（八十） (/topics/2010)
- 【有偿】求Go语言大神帮忙一个小问题 (/topics/1977)
- go接收到消息如何根据消息头判断接收 (/topics/1976)
- 使用go测试MySQL并发 不好使 (/topics/1975)
- 《The Go Programming Language》3.3节 习题 (/topics/2009)
- beego1.6.1升级到1.7版本遇到的session问题 (/topics/2000)
- golang能实现多进程吗（不是多线程，也不是协程） (/topics/1999)
- 一个简单易用的微信 sdk (/topics/1998)
- 官方文档标准库中文版页面显示异常 (/topics/1997)
- Golang请求阿里云api服务器 (/topics/1996)



控制台



软件工程师薪水









代缴社保公司



编程学习网站

🔥 开源项目（/projects）

-  (/p/go-wechat-sdk)
Golang 开发的微信 SDK go
-  (/p/leaps-golang)
Leaps(golang) (/p/leaps-;
-  (/p/gabs)
Go 的 JSON 处理库 gabs (
-  (/p/graph)
Go 语言的广义图形包 grapl
-  (/p/goraph)
实现图形数据结构和算法 g



关于 (/wiki/about) | API (/api) | 贡献者 (/wiki/contributors) | 帮助推广 (/wiki) | 反馈 (/topics/node/16) | Github (http://github.com/studygolang) | 新浪微博 (http://weibo.com/studygolang) | 内嵌Wide (/wide/playground) | 免责声明 (/wiki/duty)

©2013-2016 studygolang.com 采用 Go语言 (http://golang.org) + MYSQL 构建 (http://www.mysql.com/) 当前在线：30人 历史最高：300人 运行时间：233h25m26.191568478s

网站编译信息 版本：V2.0.0/master-32141279e63198c1a1c2a3fbc086d5fac85adb0c，时间：2016-09-07 23:52:45.083794125 +0800 CST

Go语言中文网，中国 Golang 社区，致力于构建完善的 Golang 中文社区，Go语言爱好者的学习家园。京ICP备14030343号-1



(http://www.ucai.cn?fr=studygolang)



(http://click.aliyun.com/m/4526/)



(https://portal.qiniu.com/signup?code=3lfz4at7pxfma)