

# ios 开发 Xcode 环境基础

冷武橘

## 一、环境变量

### 1、常见环境变量

`$(PRODUCT_BUNDLE_IDENTIFIER)`

- product bundle identifier 就是 App 的唯一标识

`$(PRODUCT_NAME):product name // 产品名字`


- PRODUCT\_NAME 就是可以看做是 App 的名字，App 的名称可以在，也可以直接修改 Info.plist 中对应的键，还可以在 Build Settings 中修改 Product Name 来修改（默认是 \$(TARGET\_NAME):即当前选中的 target 对应的值

`${EXECUTABLE_NAME}`

- Executable name – 执行程序名，默认与 PRODUCT\_NAME 一致。

`$(INHERITED)`

- 继承

在 Xcode 项目中很多地方都有  (inherited) 的意思就是继承上一级或依赖项的配置。通过 CocoaPods 集成

的项目，\$(inherited) 将会包含 Pods.xcodeproj 中的配置。

`$(SRCROOT)`

- 项目的根目录，

`SRCROOT=/Users/hanlitao/Desktop/demo`

`$(PROJECT_DIR) = $(SRCROOT)`

- 在实际项目中,PROJECT\_DIR 和\$(SRCROOT) 可以相互替换使用。

`${PODS_ROOT}`

- 代表的是 pod 目录

## 2、Bundle name 和 Bundle display name

Bundle name：简单来说就是 App 安装到 iOS 机子里的 App 文件夹名。也就是 ipa 的包名

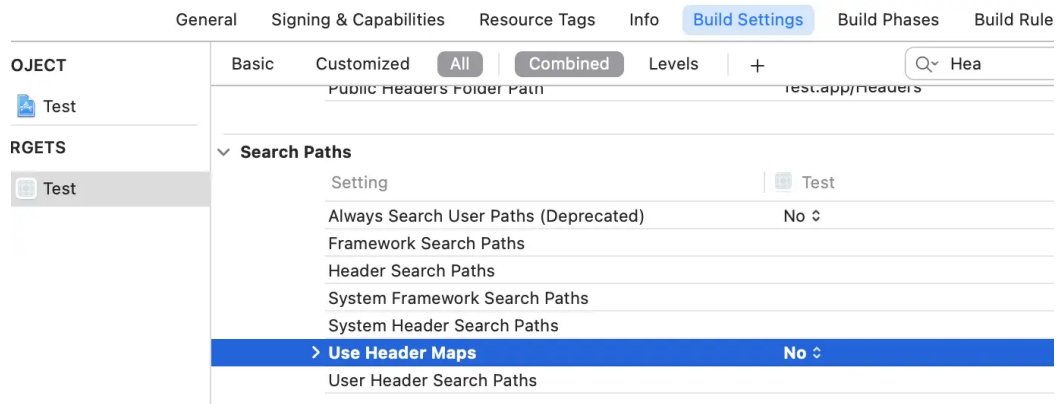
Bundle display name：app 在设备上的显示名字

## 二、Search Paths

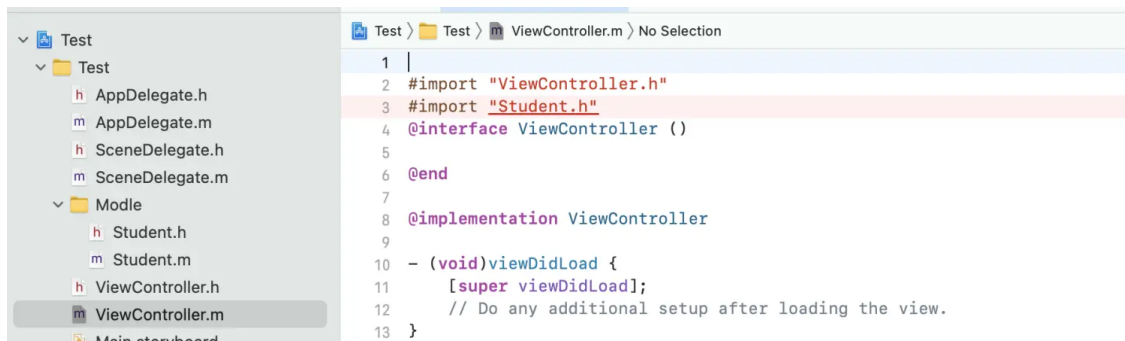
- Use Header Maps

默认值是 Yes,开启这个开关后，在本地会根据当前目录生成一份文件名和相对路径的映射，依靠这个映射，我

们可以直接import工程里的文件，不需要依靠header search path。如果将Use Header Maps设置为NO,则不会生成映射。



截屏 2021-08-20 下午 3.40.26.png



截屏 2021-08-20 下午 3.42.47.png

如图这样将 Use Header Maps 设置为 NO，直接#import "Student.h"会编译报错。这是因为上面所说的原因没有生成对应的路径映射。此时，Student.h和 ViewController.h是在同一目录下，直接导入Student.h的路径是不正确的。Modle/Student才是正确的路径。

```
1
2 #import "ViewController.h"
3 #import "Modle/Student.h"
4 @interface ViewController ()
5
6 @end
7
8 @implementation ViewController
9
10 - (void)viewDidLoad {
11     [super viewDidLoad];
12     // Do any additional setup after loading the view.
13     Student *s = [[Student alloc] init];
14 }
15
```

截屏 2021-08-20 下午 3.49.37.png

这样#import "Modle/Student.h"才会是正确的路径。总结一下：  
如果将Use Header Maps 设置为NO,不会生成映射路径，你需要导入自己正确的文件路径才行

当然你直接导入绝对路径也能解决问题



image.png

- Use Header Search Paths: 用户头文件的搜索目录源
- Header Search Paths: 本地头文件的搜索目录源

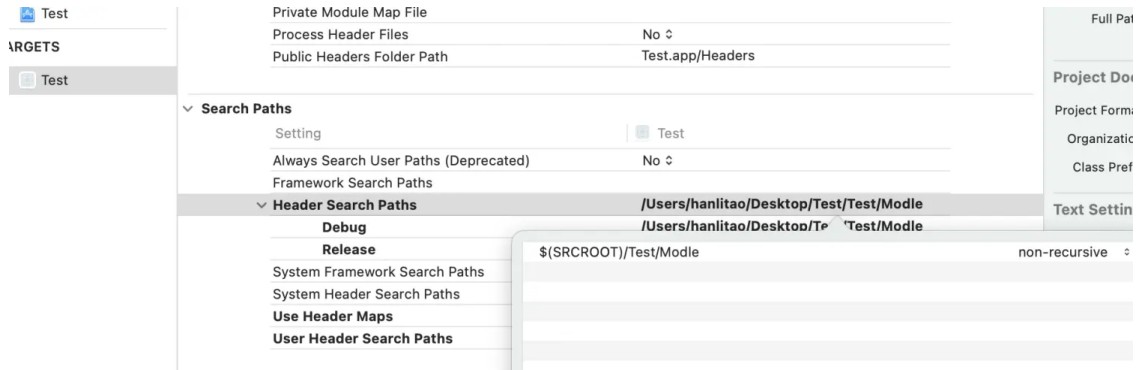
怎么理解呢？

首先明确的是 Use Header Search Paths 和 Header Search Paths 还是和导入头文件有关，再回忆一下导入文件的方式有 `<>` 和 `""` 两种。`#import <>` 和 `#import ""` 的区别是什么：在开发中我们的经验是系统库和第三库用 `<>`，自定义文件用 `""`。

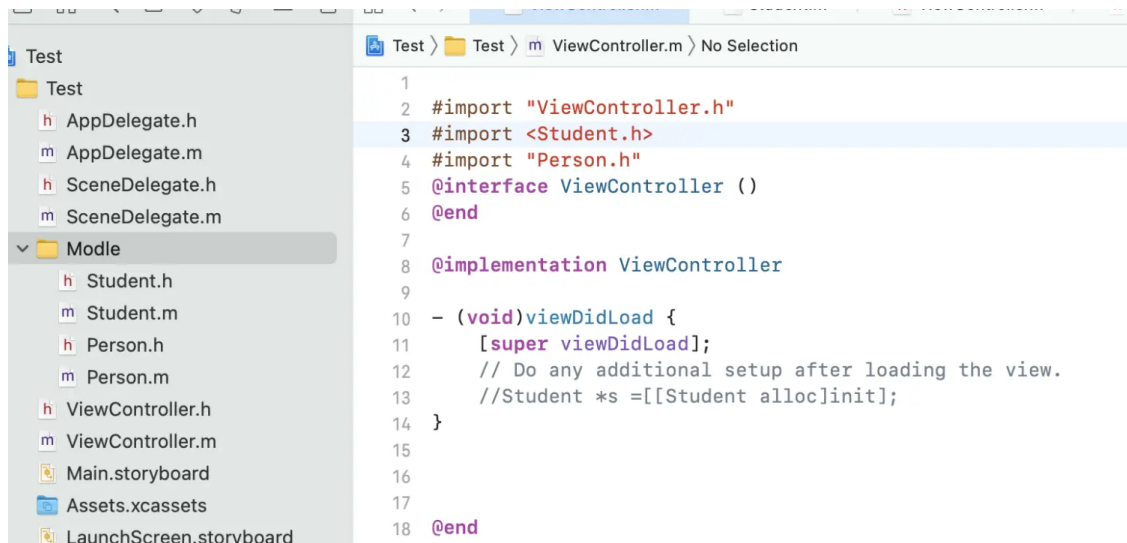
而更详细具体的来讲：`<>` 是从系统目录空间（对应 Header Search Paths）中搜索文件，而 `""` 是先从用户目录空间（对应 User Header Search Paths）中搜索文件，如果搜索不到再从系统目录空间中搜索文件。

因此 Use Header Search Paths 和 Header Search Paths 的作用相

同的，区别就是 Header Search Paths 指定了它们是系统目录；而 Use Header Search Path 指定了它们是用户目录。



截屏 2021-08-20 下午 5.31.09.png

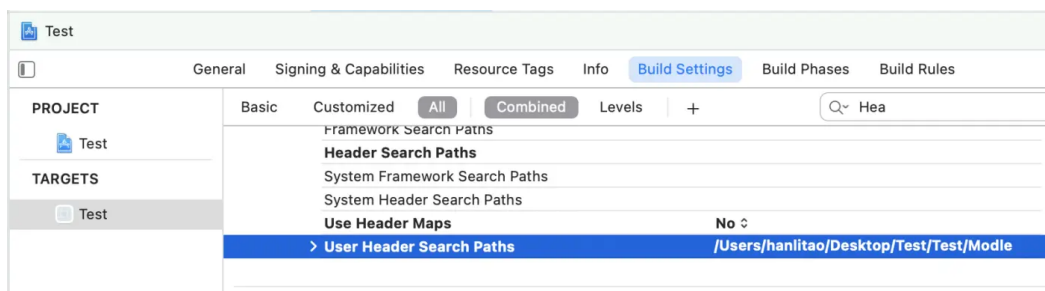


截屏 2021-08-20 下午 5.36.09.png

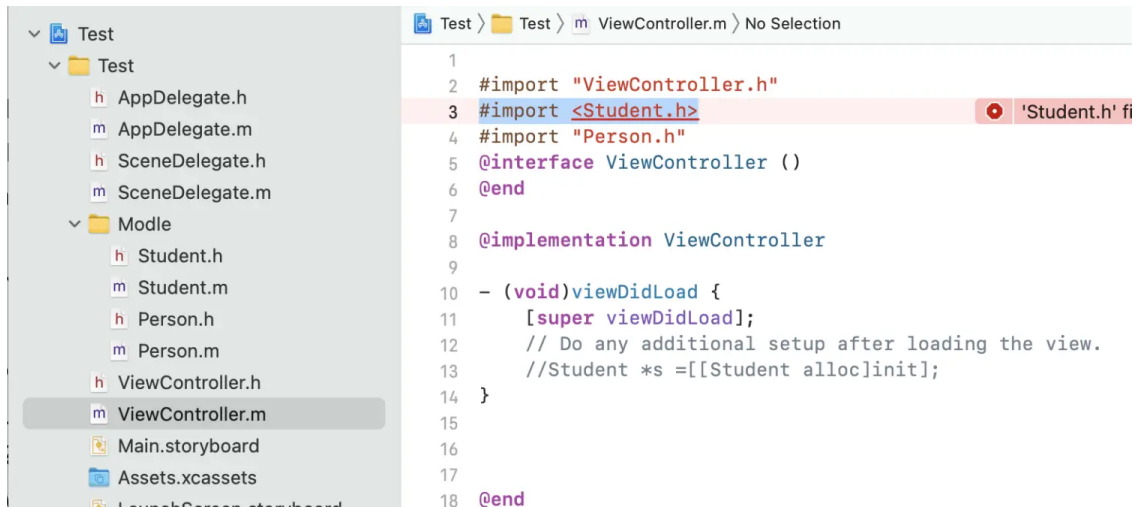
- 1、如果指定了 Header Search Paths 的目录源，这样直接导入文件就可以了。
- 2、当 #import "Person.h" 时，会先去 Use Header

Search Paths 的目录源开始找，而 Use Header Search Path 此时没有目录源，就会从接着从 Header Search Paths 的目录源开始查找。

- 3、当 `#import <Student.h>` 时，就会直接从 Header Search Paths 的目录源开始查找。



截屏 2021-08-20 下午 5.41.57.png



截屏 2021-08-20 下午 5.42.25.png

- 当 `#import <Student.h>` 时，只会直接从 Header Search Paths 的目录源开始查找。而此时 Header Search Paths 找不到，自然编译就会报错

## 小结一下：

- 1、`import ""` 在 Use Header Search Paths 和 Header Search Paths 中都可以使用。
- 2、`import <>` 在 Header Search Paths 中或者系统库时才能使用。
- 3、`import ""` 导入头文件查找的顺序应该是：  
如果有路径映射（Use Header Maps=YES），直接通过映射表获取；  
没有路径映射（Use Header Maps=NO），会先去 Use Header Search Paths 的目录源开始找，如果找不到再去 Header Search Paths 中查找。



- 4、import<>导入头文件，可以从 Header Search Paths 查找。所以使用 import<>不仅仅是引用系统类，还可以是自定义类。