

使用 Jenkins 进行自动打包，打包生成的文件格式为 xxx.ipa。自动打包的设置原理是：通过 jenkins 从 svn 服务器将最新的项目源代码下载到指定的目录，然后再进行打包，若有需要可以上传到蒲公英等第三方托管平台。

注意事项：

（1）打包时区分是开发测试使用的 ipa 包，还是需要上传到 appStore 的 ipa 包；开发测试环境时，使用开发证书和开发描述文件，而发布环境时使用发布证书和发布描述文件。

（2）必须启用 jenkins 时才能打开浏览器。

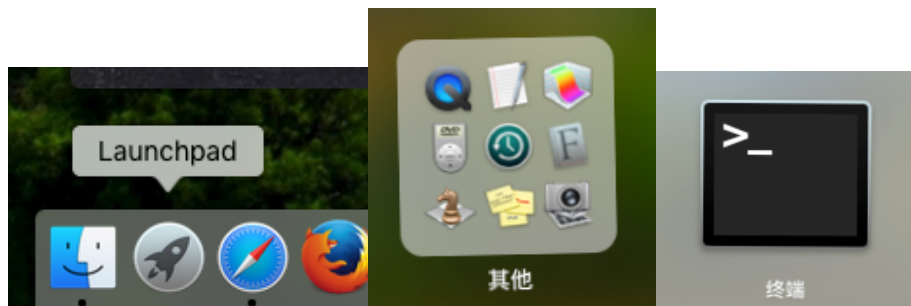
具体步骤如下：

1、jenkins 的安装、配置、使用介绍

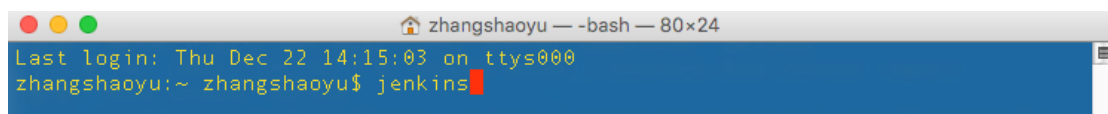
<http://blog.csdn.net/potato512/article/details/52289136>

2、jenkins 的启用

打开终端：Dock—Launchpad—其他-终端



终端命令：jenkins



3、jenkins 使用（打开浏览器前必须启用 jenkins，localhost 为 IP 地址）

浏览器地址：<http://localhost:8080>

4、项目配置

该项目配置是通过 jenkins 从 svn 下载最新的源代码，然后根据项目是开发环境（开发证书、开发描述文件），还是发布环境（发布证书、发布描述文件）进行打包。基本所需要的配置如下：

(1) General

项目名称、描述、勾选丢弃的构建

(2) 源码管理

勾选 None

(3) 构建触发器

无

(4) 构建环境

无

(5) 构建

选择 Execute shell

Command

#<----->

# 删除旧源码目录->新建源码目录->从 svn 导出最新代码->清理旧文件->清除旧项目->打包->上传

pwd

# 名称配置

checkout\_name="checkout" #项目源码下载存储目录

project\_name="VSTMALL" #项目名称

# 配置项目版本

# targetProject\_sdk="iphoneos8.0"

targetProject\_destination="generic/platform=iOS"

configuration="Release"

scheme="\$project\_name"

workspace\_name="\${project\_name}.xcworkspace"

# 目录配置

save\_path="/Users/zhangshaoyu/Desktop/uploadIPA"

archive\_path="\$save\_path/\${project\_name}.xcarchive"

ipa\_path="\$save\_path/\${project\_name}.ipa"

log\_path="\$save\_path/log.txt"

# svn 配置 (SVN 项目源码地址、源码下载保存目录、帐号、密码)

```
svn_path="http://192.168.32.211:8201/svn/VSTMALL/branches/copyright/iOS/VSTSUPPORT"
checkout_path="$save_path/$checkout_name"
svn_name="zhangshaoyu"
svn_password="123456"
# 配置签名证书、描述文件
# 开发
#codeSignIdentity="iPhone Developer: shaoyu zhang (5AB779CDEF)"
#provision_UUID="a0487328-bcd7-405e-f626-39fgggg09a52"
#provisioning_profile="developProfile"
# 发布
codeSignIdentity="iPhone Distribution: shaoyu Technology (zhang) Co., Ltd."
provision_UUID="06879a23-8b9c-432d-ef89-6gh5ij925kl6"
provisioning_profile="distributionProfile"
# 配置蒲公英
upload_path="$save_path/${project_name}.ipa"
pgy_userKey="a512b58c56285d23456f011bfc706509"
pgy_apiKey="ab9c240d2efb9cf17d9642f3b5ccca5a"
```

```
echo "正在删除旧源码"
# 删除旧源码目录
rm -rf "$checkout_path" >> $log_path
```

```
echo "正在创建新的源码目录"
# 新建源码目录
cd "$save_path" >> $log_path
pwd
mkdir "$checkout_name" >> $log_path
```

```
echo "正在从 svn 下载最新的源码"
# 从 svn 导出最新代码
svn checkout "$svn_path" "$checkout_path" --username "$svn_name" --password
```

```
"$svn_password" >> $log_path
```

```
echo "正在删除旧文件"
```

```
# 删除旧文件
```

```
rm -rf "$log_path" >> $log_path
```

```
rm -rf "$archive_path" >> $log_path
```

```
rm -rf "$ipa_path" >> $log_path
```

```
echo "正在清除构建项目缓存"
```

```
# 重要, 执行 xcodebuild 命令时, 必须进入项目目录
```

```
cd "$checkout_path" >> $log_path
```

```
pwd
```

```
# 清理构建目录
```

```
xcodebuild clean -configuration "$configuration" -alltargets >> $log_path
```

```
echo "正在打包"
```

```
# 归档 (其他参数不指定的话, 默认用的是.xcworkspace 或.xcodeproj 文件里的配置)
```

```
xcodebuild archive -workspace "$workspace_name" -scheme "$scheme"  
-destination "$targetProject_destination" -configuration "$configuration"  
-archivePath "$archive_path" CODE_SIGN_IDENTITY="$codeSignIdentity"  
PROVISIONING_PROFILE="$provision_UUID" >> $log_path
```

```
echo "正在导出 ipa 包"
```

```
# 导出 IPA
```

```
xcodebuild -exportArchive -exportFormat IPA -archivePath "$archive_path"  
-exportPath "$ipa_path" -exportProvisioningProfile "$provisioning_profile" >>  
$log_path
```

```
echo "正在上传 ipa 到蒲公英"
```

```
# 上传 IPA 到蒲公英
```

```
# curl -F "file=@$upload_path" -F "uKey=$pgy_userKey" -F  
"_api_key=$pgy_apiKey" https://www.pgyer.com/apiv1/app/upload
```

#<----->

## （6）构建后操作

### **Archive the artifacts**

用于存档的文件，即保存打包的文件存储路径