

目录	1
前言	2
简介	3
Release	5
Packages	8
签名 Release (Release.gpg)	15
密钥和公匙	17
附录	22

本文由威锋网技术组出品，仅供爱好者交流使用，不得用于商业用途，内容如侵犯到您的权益，请与本站联系。
本文内容谢绝未经授权转载，如需转载请联系本站
✉ Qingxing@office.weiphone.com
© WEIPHONE TECH TEAM

本帖中任何涉及文本编辑的内容，必须保证文本为 UTF-8，Unix 编码，Unix/Linux 换行符(LF)。

Linux 系统下默认就是符合标准的，Mac OS X 下可能需要调整编码，iPhone OS 下编辑取决于软件，WinSCP 自带的编辑器是符合规范的，但在默认设置下，如果文本中有中文，那么文本格式就不是 UTF-8，而是 GB2312。

Windows 下记事本/写字板等文件编辑器，默认保存格式都是 ANSI+Windows 编码，所以需要转换编码。可以使用 UltraEdit 之类的工具，我个人觉得 Visual Studio 的编辑器也不错，在高级保存选项中设置一下就可以了。

简介

Cydia 中的软件源地址保存在 `/etc/apt/sources.list.d/` 目录下，分为两种：

1. 以软件包形式安装的软件源，如 Cydia 默认的软件源，Modmyi、BigBoss 等，该类软件源是以独立的.list 文件记录软件源地址。
2. 由用户在 Cydia 里输入地址添加的软件源，该类软件源全部保存在 `cydia.list` 中。

以上两种形式的软件源本质没有差别，但只软件源的文件存放位置上是有不同的。

以 Saurik 自己的软件源为例。打开 `saurik.list` 文件，可以看到有以下内容（以#开头的行属于注释，没有实际用途）

```
deb http://apt.saurik.com/ tangelo-3.7 main
```

那么，软件源的 Release 文件的地址是 <http://apt.saurik.com/dists/tangelo-3.7/Release>

Cydia 就是通过或许 Release 文件进而检测软件源的存在，然后再下载记录软件包相关信息的 Packages 文件。

而对于在 Cydia 中输入软件源地址进行添加的这种源，方法有些不同。因为默认情况下，输入地址以后，Cydia 会认为 Release 文件存放于输入的地址所在的目录下。在 list 文件中的记录信息为：

```
deb http://xxx.xxx.com/ ./
```

并不会继续往子目录里检测 Release 文件。

因此，搭建 Cydia 软件源首先需要配置服务器上相关目录和文件的存放位置。

取决于软件源的添加方式，文件/目录的路径设置可以分为以下两种情况：

1. 只公布软件源的地址，让用户手动输入地址来添加。这种方式的文件/目录设置最为简单。

以威锋源为例，威锋源的地址为 <http://app.weiphone.com/cydia>

那么在郁闷 `app.weiphone.com` 下，存在目录 `cydia`。目录 `cydia` 即使软件源所在的目录。

将文件 Release 放置在 `cydia` 目录下即完成了软件源的搭建。

要注意，搭建软件源，必须保证至少有 Release 和 Packages 两个文件。

2. 以软件包的形式安装，即直接在 /etc/apt/source.list.d/ 目录以文件的形式添加，那么这个就没有什么固定的位置了。

基本上仿照 Saurik 和 BigBoss 的 list 文件的格式即可。

list 文件格式：

```
# do not edit this file to add your own custom sources

# this file is subject to be upgraded as part of a package

# either add your own .list file to /etc/apt/sources.list.d

# or add your entry to the global /etc/apt/sources.list file

deb 软件源地址 项目名 类型
```

文件名任意

其中，Release 文件应放置于 软件源地址/项目名/ 这个目录下，Packages 文件放置位置由 Release 文件来决定（详情见后文）。

之前提到过，搭建软件源必须要有 Release 和 Packages 两个文件。前者的作用是记录软件源本身的相关信息，而后的作用则是记录具体软件包的存放位置和安装信息等数据。

Release 文件的格式

```
Origin: BigBoss
Label: BigBoss
Suite: stable
Version: 1.0
Codename: BigBoss
Support: http://cydia.saurik.com/support/*
Architectures: darwin-arm iphoneos-arm
Components: main
Description: Apps & Themes! We host your apps. Email me.
MD5Sum:
be8806290d5904cdf45b542706f6a3ad 165020 main/binary-darwin-arm/Packages
03026ac993187b0eeca50466f64fb3c 35049 main/binary-darwin-arm/Packages.gz
93091f5ca485e066d5fc32ca1327830c 3031061 main/binary-iphoneos-arm/Packages
d7aec18dee5b627339b77a9d91e3f92f 368222 main/binary-iphoneos-arm/Packages.bz2
```

其中红字部分是必填内容，紫色部分是可选内容。

每项的作用：

必需

Origin: 软件源名称，可以使用中文（Cydia 的软件源列表中显示的标题）

Label: 同上，也可以使用中文

Suite: 软件源的类型，比如正式源，测试源等，可以分别用 stable, beta, unstable 等来表示，一般填 **stable** 就可以了

Version: 版本号，这个其实不重要，随便填，一般都是写 1.0

Codename: 代码代号，比如 BigBoss 的就写 BigBoss，威锋的就写 WeiPhone，也没什么限制，只能用英文

Architectures: 结构。iPhone 平台统一写 **iphoneos-arm**

Components: **main**

Description: 软件介绍，可以使用中文和 html 代码，具体能使用哪些代码在下面会介绍。

可选

Support: 支持，没什么作用，除非特别需要，否则可以不要这个。

MD5Sum: 不是必须的，但如果 Packages 文件位置不与 Release 文件在同一目录下，则必须有此项。另外，如果需要签名 Release 文件，也必须有这个。关于 MD5Sum 的格式，在下文也会介绍。

Description 格式

显示在 Cydia 中每个软件页面最下方。

不能直接换行，如果要实现显示换行显示，可以使用
代码。

要加粗显示，可以使用代码

可以使用 html 代码设置字体颜色。

不能使用超链接代码。

代码：**Description: WeiPhone-威锋网为您提供 iPhone 所需软件/补丁。

联系我们: weip.com@ gmail.com**

实际显示效果如下：



MD5Sum 格式

MD5Sum 是用来记录 Packages 即相关文件的文件大小，存放路径和 MD5 值。

MD5Sum: 之后另起一行，开头要空一格（论坛里用 quote 或 code 代码之后都不显示开头的空格）

每行格式是：

空格+文件 MD5 值+空格+文件大小(以字节为单位)+文件路径（全部没有+）

而文件路径的格式则有文件存放位置来决定。

如果是没有设置 Release 文件的路径，那么 Release 和 Packages 同在根目录，则路径为 Packages，直接就是文件名，没有其它路径。

如果在之前设置了 Release 文件的路径，如 Saurik 和 BigBoss，那么 Packages 的路径为 main/binary-iphoneos-arm/Packages，起始文件夹，即 main，要与 Release 文件同在一个目录下。main 和 binary-iphoneos-arm 要与 Releases 中的对应信息相匹配。

对于 Packages.gz 和 Packages.bz2，格式相同，也记得要另起一行。

因此，Packages 文件的保存位置就是 软件源地址/dists/main/Codename（与 Release 文件中一致）/main/binary-iphoneos-arm 下。

Release 文件的示例可以参看以下几个地址：

<http://app.weiphone.com/cydia/Release>

<http://apt.saurik.com/dists/tangelo-3.7/Release>

<http://apt.bigboss.us.com/repofiles/cydia/dists/stable/Release>

生成 Packages 文件的方法

Packages 文件格式

```
Package: 3proxy
Version: 0.5.3k-2
Architecture: iphoneos-arm
Maintainer: Jay Freeman (saurik) <saurik@saurik.com>
Installed-Size: 1208
Filename: debs/3proxy_0.5.3k-2_iphoneos-arm.deb
Size: 416318
MD5sum: f84cb6764a9df30ab326c17f51de15fd
Section: Networking
Priority: optional
Homepage: http://3proxy.ru/download/
Description: tiny free proxy server
Name: 3proxy
Depiction: http://cydia.saurik.com/info/3proxy/
Tag: purpose::daemon, role::hacker
```

Packages 实质上是 deb 文件的 control 信息集合，另外多了 deb 文件本身的信息。
(control 文件是记录软件包本身的信息，但没有 deb 文件本身)

多出的 deb 文件信息为：

- Filename (deb 文件位置与文件名)
- Size (deb 文件大小，字节为单位)
- MD5sum (deb 文件 MD5 值)

要保证 Cydia 能正确获取 deb 的相关数据，除了以上 3 个 deb 文件信息以为，还必须保证至

少有以下两项：

- Package (软件标识符，相当于身份证号码，用于识别唯一的软件)
- Version (软件版本号)

这里重点关注的是 Filename。Filename 项决定了在服务器上 deb 文件的存放位置。如果没有上级目录，即 deb 文件和 Packages 文件处于同一目录，那么只需要一个单独的文件名就可以了，不需要加上目录名，即 xxxx.deb。但这样在软件数量多的时候会导致管理上的混乱。

假设 deb 文件存放在 debs 目录下，那么 Filename 就应该是 debs/xxx.deb 这样的格式。

其余项请参与 deb 制作教程。

一个软件包的全部信息在 Packages 文件中是以连续的一段存在，每一行是不同的信息。不同软件之间以空行做分隔。

生成 Packages 文件

方法一

环境：任何支持 dpkg 命令的 unix 系统

工具：dpkg 及相关组件。

- iPhone OS (已越狱)：自带 dpkg-deb 命令，无 dpkg-scanpackages 命令
- Debian/Ubuntu：自带 dpkg-deb 和 dpkg-scanpackages 命令
- Fedora: 无 dpkg 相关命令
- Mac OS X：无任何 dpkg 相关命令
- Windows：非 unix 系统，无 dpkg 相关命令

以上系统中，Fedora 不支持 dpkg，也无法通过安装系统组件来支持 dpkg，因此 Fedora 系统不能用于生成 Packages 文件。

而对于 Mac OS X 系统，本身虽然不支持 dpkg，但可以通过安装一个叫 Fink 的软件来获得 dpkg 命令。

Fink 下载页面：<http://www.finkproject.org/download/>

如何安装请见官网说明。

对于 Windows 系统，可以通过安装 Cygwin + dpkg 来获取 dpkg 命令。

Cygwin 下载地址：<http://www.cygwin.com/setup.exe>

dpkg 命令可以直接在 Cygwin 中获取。

但个人不推荐在 Windows 下使用 dpkg，因为 Cygwin 体积很大，与其安装这个，还不如直接安装个 Ubuntu 来的简单。

各个系统下生成 Packages 的方法基本一样。

最简单的方法是利用 dpkg-scanpackages 命令来制作 Packages 文件。

iPhone OS 不适用（因为没有这个命令）

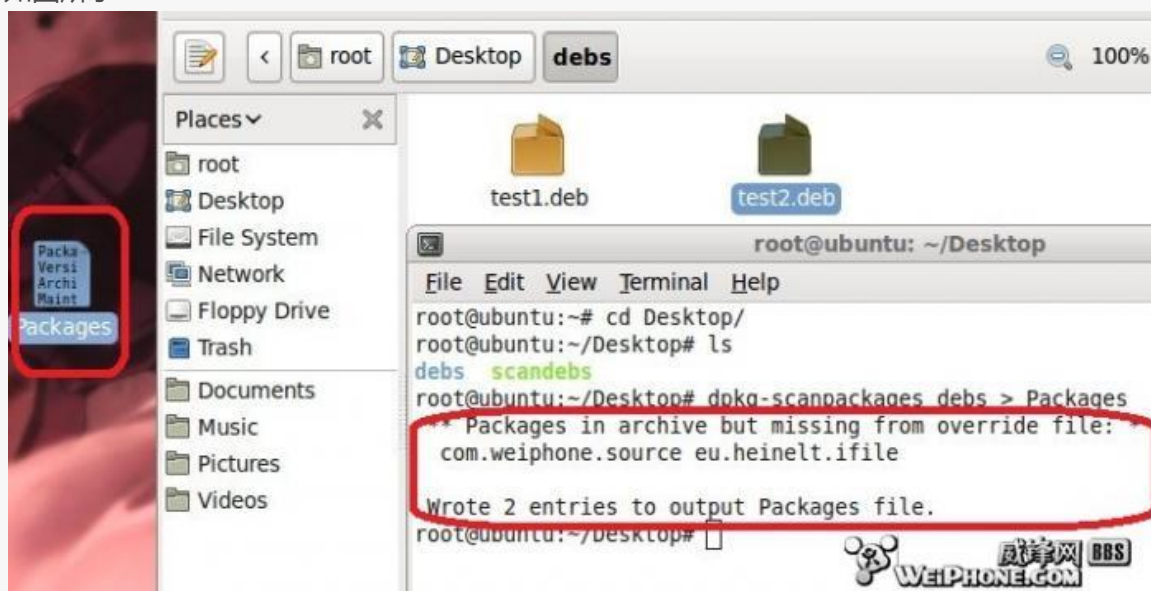
1. 将需要发布的 deb 文件放在一个目录，比如说 debs 下
2. 命令行里进入到 debs 目录的上级目录
3. 执行命令

```
dpkg-scanpackages debs > Packages
```

，其中 debs 为 deb 存放目录。运行该命令可能需要 root 权限，可以使用 root 帐户登录，login root 或 su root 或使用 sudo

4. Packages 存放于当前目录

如图所示



如果指定的文件夹不存在，则提示 dpkg-scanpackages: error: Binary dir debs not found

如果在指定的文件夹内没有 deb 文件，则提示 Wrote 0 entries to output Packages file.

注意：Debian/Ubuntu(9.10 及更早)系统下的 dpkg-scanpackages 命令不会输出 Name, Author, Sponsor, Depiction 等信息不会自动添加进去，所以需要手动加入这些数据。Mac OS X 和 Ubuntu 10.04 下无此问题。

以 dpkg-scanpackages 命令来生成 Packages 文件有一个很大的限制，就是必须保留全部要发布的和已经发布的 deb 文件，在 deb 数量多的情况下会浪费大量磁盘空间。

因此可以使用改进命令：

```
dpkg-scanpackages debs >> Packages
```

，这样不会每次都重新生成 Packages 文件，而是将 debs 目录下的 deb 信息添加到已有的 Packages 文件末尾。在新增软件的时候没什么问题，但如果是软件更新，那么就会出现重复条目的问题，这时候需要手动删除旧的软件包信息。

方法二

上面的方法里也提到过，可以直接编辑 Packages 文件，因此用文本编辑器将相关数据写入 Packages 也是可以的，只是这样工作量会比较大，但这种方法没有系统的限制，任何系统下，不管有没有 dpkg 相关命令都可以使用。格式也是一样的，每项之间只要把 Packages 放在第一个就可以了，其余项没有顺序要求。

由于绝大部分数据可以从 control 里直接复制，需要自己获取的就只有 Filename, Size 和 MD5sum。

在 Unix 系统下，包括 Mac OS X 和 Linux，以及 iPhone OS，获取文件大小可以使用命令 ls -l；获取 MD5 值可以使用命令 md5sum

另外，在 Windows 下要解包 deb 文件可以使用 7-zip，control 文件位于 control.tar.gz 下

方法三

生成 Packages 和 Release 文件只是最基础的步骤，实际还需要压缩 Packages 文件，有时候还需要签名 Release 文件，为了简化操作，我自己写了一个脚本，具有以下功能：

自动从服务器上获取 Packages 文件，不需要每次都事先准备

- 自动生成 Release 文件
- 自动生成 Packages, Packages.bz2 和 Packages.gz

- 在生成 Packages 文件之后无需保留 deb 文件
- 保留 control 里的全部信息
- 不会出现重复条目
- 自动签名 Release 文件

下载地址：<http://app.weiphone.com/files/scandebbs.tar.gz>

使用之前需要先修改下脚本（只需修改一次即可）【因为更新过，所以图片所示的行数已不匹配了，但是内容基本还是一样的。】

1. 第 46 行，将 wget 后的地址改为自己源的 Packages 的地址


```
18 #
19 # other repo should modify this url
20     wget http://app.weiphone.com/cydia/Packages
21 # end of modify Packages' url
22 #
23 fi|
24 #
```



另外，因为 iPhone OS 本身没有 wget 这个命令，可以去 Cydia 安装一个 wget。使用此命令需要联网，否则请事先准备好 Packages 文件。

2. 第 132, 136, 140 这三行，将 Packages, Packages.bz2 和 Packages.gz 根据需要自己修改（如果需要指定这几个文件的存放位置）

```
87 echo " $md5 $size Packages" >> rel
88 md5sum Packages.bz2 > md5tmp
89 md5=$(cut -c -32 md5tmp)
90 size=$(stat -c %s Packages.bz2)
91 echo " $md5 $size Packages.bz2" >> rel|
92 md5sum Packages.gz > md5tmp
93 md5=$(cut -c -32 md5tmp)
94 size=$(stat -c %s Packages)
95 echo " $md5 $size Packages.gz" >> rel
```



3. 第 151-158 行，引号内的内容根据需要自己来修改（此部分用于生成 Release 文件）

```
97 echo "Origin: WeiPhone-威锋网" > rl
98 echo "Label: WeiPhone-威锋网" >> rl
99 echo "Suite: stable" >> rl
100 echo "Version: 1.0" >> rl
101 echo "Codename: WeiPhone" >> rl
102 echo "Architectures: iphoneos-arm" >> rl
103 echo "Components: main" >> rl
104 echo "Description: WeiPhone-威锋网为您提供iPhone所需软件/补丁。<br><br>联系我们: <strong>weip.com@gmail.com</strong>" >> rl
```



4. 第 166 行，引号内的内容修改为自己的私匙的用户名和密码

```
109 #  
110 # sign Release  
111 gpg -abs -r "HuangRui" -o Release.gpg Release  
112 # end of signing Release
```



5. **Ubuntu 用户请注意：**Ubuntu 下的 GPG 默认开启了 use-agent 功能，所以在签名 Release 的时候会弹出一个提示框，而这个提示框根本就没用，所以要禁用。
6. **禁用方法：**打开 ~/.gnupg/gpg.conf（~表示当前用户）或者直接用命令

```
gedit ~/.gnupg/gpg.conf
```

搜索 use-agent，将其注释掉（即修改为 #use-agent）

使用方法：

将 scandebs 文件放置在任意位置

1. 准备 debs 文件夹，需在 scandebs 同级目录下
2. 将要更新的 deb 文件放置在 debs 目录下
3. 【可选】将 Packages 文件放置在 scandebs 同级目录下（如果需要修改 Packages 文件中的已有数据）
4. 以命令 sh scandebs + 参数 来执行脚本；或 将文件属性设为 0755，命令 chmod 0755 scandebs，之后可以直接使用命令 ./scandebs + 参数
5. 【可选】如果提示要你输入签名 Release 文件的密码，输入之
6. 得到 Packages，Packages.bz2，Packages.gz，Release 和 Release.gpg 【可选】这 5 或 4 个文件

显示帮助

```
./scandebs -h
```

或

```
./scandebs --help
```

显示版本号

```
./scandebs -v
```

或

```
./scandebs --version
```

不签名 Release (无参数)

```
./scandeb
```

使用签名 Release 功能

```
./scandeb -s
```

或

```
./scandeb --sign
```

手动修改 Packages 注意事项

前面说过，Packages 文件可以手动编辑，有时候仅仅需要更改一些字段的内容，如果因为这个要重新打包 deb，不一定合算，因此手动编辑 Packages 可以省下一些工作量。以下是一些注意事项：

- 以下字段不可修改：Package，Version，Size，MD5Sum，Depends，Pre-Depends，Provides，Conflicts，Replaces
- 其余字段可以任意修改，无需与 deb 里的 control 一致，因为 Cydia 本身不是读取的 control 的数据，而是从 Packages 中获取数据。
- 修改字段以后，如果以 dpkg-scanpackages 重新生成 Packages，那么修改过的部分会被还原成 control 里的原始数据。

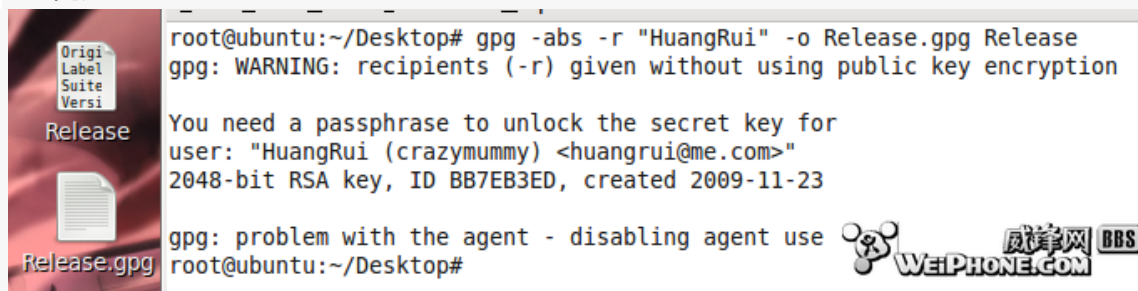
签名 Release 的方法

1. 获取用于签名的 GPG 密匙（方法见下文）
2. 运行命令

```
gpg -abs -r "HuangRui" -o Release.gpg Release
```

其中引号内的字符串为用于签名的密匙的用户名（用于区分不同的签名密匙）

3. 输入密码
4. 完成



为什么要签名 Release

虽然不签名 Release 文件也没什么大的问题，但是签名 Release 有以下几个用途：

- 解决 Cydia 刷新出现的 Wrong language code zh-Hans 或 Wrong language code zh-Hant 的问题（当然如果别的源有这个问题的话还是会报错）
- 所有源内的数据均经过验证，确保安全（见图）



上传至服务器

1. 将 Packages 文件压缩为 Packages.bz2 (压缩格式 bzip2) 和 Packages.gz (压缩格式 : gz) (如果是用的我写的脚本则可省略自己压缩的这一部)
2. 修改 Release 文件 (主要是 MD5 值和文件大小)
3. 【可选】签名 Release 文件 , 生成 Release.gpg
4. 将 Release , Release.gpg , Packages , Packages.bz2 和 Packages.gz 以及要更新的 deb 文件上传到服务器指定位置。
5. 以上所有文件的属性需设为 0755 (在 ftp 上设置)

用于签名 Release 的密匙

生成密匙

可以看看这个页面 <http://keyring.debian.org/creating-key.html>

生成密匙的操作是在命令行下完成，其中还需要有人机交互，所以 WinSCP 的命令行是不能用的，如果需要使用 iPhone 来生成密匙，请使用 Putty 或者其它 SSH 终端工具来连接。

运行命令

```
gpg --gen-key
```

提示

Please select what kind of key you want:

- (1) DSA and Elgamal (default)
- (2) DSA (sign only)
- (5) RSA (sign only)

Your selection?

一般就是 1，默认

回车后

RSA keys may be between 1024 and 4096 bits long.

What keysize do you want? (2048)

输入 1024 ~ 4096 之间的一个数值，数值越大，加密强度越高，但每次加密和解密需要的时间更多。

我这里输入的是 2048

Requested keysize is 2048 bits

Please specify how long the key should be valid.

- 0 = key does not expire
- = key expires in n days

```
w = key expires in n weeks
m = key expires in n months
y = key expires in n years
Key is valid for? (0)
Key does not expire at all
Is this correct? (y/N) y
```

密钥有效时间，一般选 0，不会过期。

问你确认，输入 y+回车（小写 y，如果要重新做则是大写 N）

Real name:

输入该密钥所有者的名称，例如 WEIP Tech Team

Email address:

密钥所有者邮件地址，自己输入

Comment :

注释，怕分不清楚的话可以加点内容进去

```
You are using the `utf-8' character set.
You selected this USER-ID:
  "WEIP Tech Team <weip.com@gmail.com>
```

继续

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit?

上面输错的话，名字错了就输入 N，加上注释就输入 C，邮件错误就输入 E，不保存退出输入 Q，确定输入 O

You need a Passphrase to protect your secret key.

设置密码，该密码在每次使用加密时都要使用，必须牢记。输入时不会明文显示，回车后会要你再输一遍以匹配。

设完密码之后就自动执行了。

注意，在结果里面有一句

```
gpg: key 0A02728F marked as ultimately trusted  
public and secret key created and signed.
```

这串字符就是所谓的密匙标识了

如果需要为这个密匙添加另外的使用者，可以用以下命令

```
gpg --edit-key 0x0A02728F
```

密匙前面要加上数字 0 和字母 x

command> 后输入

```
adduid
```

之后的操作跟前面的一样，不过在要输密码的时候只是要验证身份，不是创建密码。

最后在 command>后输入

```
save
```

假设创建了两个用户，A 和 B，原本是 A 拥有最高权限，要改为 B 拥有最高权限

执行以下命令：

```
gpg --edit-key 0x0A02728F
```

command> 后输入

```
uid 2
```

然后输入

```
Command> primary
```

输入密码后在 command>后输入

```
save
```

经过以上操作就成功生成了一个可以用于签名的密匙。

导出公匙

用于加密/签名的是刚刚申请到的私人密匙（ Private Key ），而要发布的是公匙（ Public Key ）

运行命令

```
gpg --export 0A02728F > weiphone-keyring.gpg
```

文件名自己决定

之后将生成的公匙发布就可以了。

导入公匙

需要导入公匙的是软件源的用户，下载公匙文件以后，让用户自己运行命令

```
apt-key add weiphone-keyring.gpg
```

导出私人密匙

有时候软件源不是一个人在维护，如果是多个人维护的话可能需要导出私人密匙以共其他人加密/签名。

运行命令

```
gpg --export-secret-key -a "User Name" > private.key
```

User Name 是自己在生成密匙的时候输入的用户名，最后生成的文件，文件名随意，但切记，这个密匙不要泄漏出去。

导入私人密匙

得到私人密匙之后就是导入了，运行命令

```
gpg --allow-secret-key-import --import private.key
```

密匙管理

查看公匙列表

```
apt-key list
```

或

```
gpg --list-keys
```

当密匙不再使用时，可以删除密匙。

删除公匙

```
apt-key del 0A02728F
```

或

```
gpg --delete-key "User Name"
```

查看私人密匙列表

```
gpg --list-secret-keys
```

删除私人密匙

```
gpg --delete-secret-key "User Name"
```

关于 GPG 密匙的更多命令，可以看看这个网页（英文）

<http://irtfweb.ifa.hawaii.edu/~lockhart/gpg/gpg-cs.html>

关于公匙和私匙的介绍：<http://www.hudong.com/wiki/%E5%85%AC%E5%8C%99>

关于源的搭建，也可以看看 Saurik 的文章 <http://www.saurik.com/id/7>

附录

要在 Cydia 中显示源的图标或者分类图标，只要将图标文件放到 Cydia 的目录下即可。

源的图标文件名为 域名.png，比如威锋源的地址为 <http://app.weiphone.com/cydia>，那么图标文件名应为 app.weiphone.com.png，该文件需位于 /Applications/Cydia.app/Sources/ 下，图标分辨率为 60 × 60

分类图标需位于 /Applications/Cydia.app/Ssections/ 下，文件名有一定要求。一般来说文件名跟分类名一致就可以了，但也有例外。以威锋源为例，分类 [2.x] 所对应的图标文件名应为 2.x.png，没有方括号，但其它的就要有方括号。另外，一定要注意大小写的匹配，文件名编码须为 UTF-8。

重点：

1. list 文件 /etc/apt/sources.list.d/weiphone.list

```
# do not edit this file to add your own custom sources

# this file is subject to be upgraded as part of a package

# either add your own .list file to /etc/apt/sources.list.d

# or add your entry to the global /etc/apt/sources.list file

deb http://app.weiphone.com/cydia/ ./
```

2. 公匙，其实放在哪里不重要，但 Cydia 的所有源的公匙都是放在 /usr/share/keyrings/ 下，所以尽量随主流吧

3. 安装公匙的脚本 postinst 或者 extrainst_都行，只不过通常都用 extrainst_

```
#!/bin/sh

if [[ $1 == install || $1 == upgrade ]]; then

    /usr/bin/apt-key add /usr/share/keyrings/weiphone-keyring.gpg

fi
```

记得修改文件名

4. 卸载时删除公匙的脚本，prerm 或者 postrm 都行

```
#!/bin/sh

if [[ $1 == remove ]]; then

    /usr/bin/apt-key del BB7EB3ED

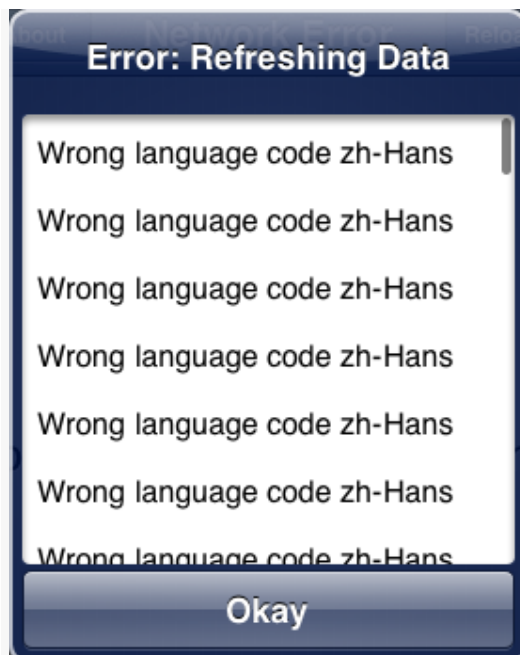
fi
```

服务器上文件存放位置，最简单的

cydia--

```
|--debs--*.deb
|--Packages
|--Packages.bz2
|--Packages.gz
|--Release
|--Release.gpg
```

Cydia Wrong language code zh-Hans 或 Wrong language code zh-Hant 的解释



其实这个不是错误，而是 Cydia 本身对中文的支持有问题。

而且，这个提示并非是错误提示，实际是属于警告的性质，因为 Cydia 本身在中文支持上有些问题，但不影响使用，所以会出现警告。

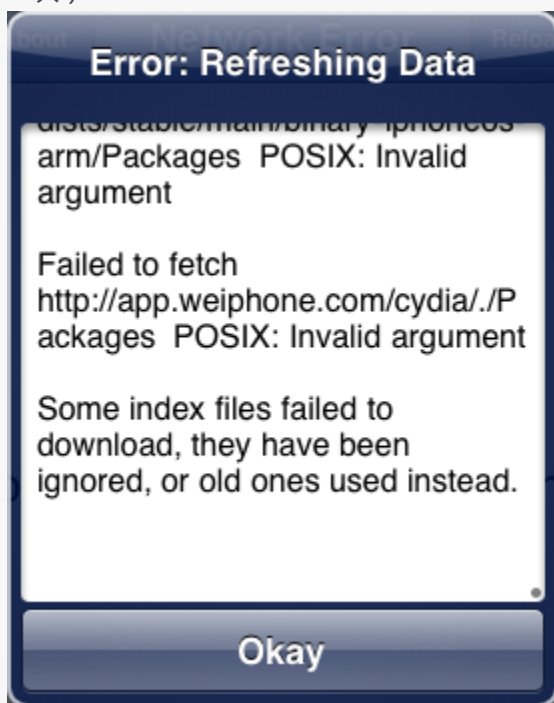
该警告的出现与 Cydia 软件源中是否存在中文无关，而是和系统语言设置有关，该问题只出现在简体中文和繁体中文系统下（不信把系统语言改成非中文的看还会出现错误）。

何时会出现错误提示：

很简单，Cydia 刷新出了问题。

一般有两种可能：

1. 网络问题导致 Cydia 不能获取 Packages 或 Release 等文件
2. Packages 和 Release 文件本身有问题，比如说格式错误或者编码错误（BigBoss 就闹过一次）



如果网络通畅的话就不会出现这个错误提示了。

DEB 制作教程： http://app.weiphone.com/files/how_to_make_deb.pdf