

## 用dpkg命令制作deb包方法总结

如何制作Deb包和相应的软件仓库，其实这个很简单。这里推荐使用dpkg来进行deb包的创建、编辑和制作。

首先了解一下deb包的文件结构：

deb 软件包里面的结构：它具有DEBIAN和软件具体安装目录（如etc, usr, opt, tmp等）。在DEBIAN目录中起码具有control文件，其次还可能具有postinst(postinstallation)、postrm(postremove)、preinst(preinstallation)、prerm(preremove)、copyright (版权)、 changlog （修订记录）和conffiles等。

control: 这个文件主要描述软件包的名称（Package），版本（Version）以及描述（Description）等，是deb包必须具备的描述性文件，以便于软件的安装管理和索引。同时为了能将软件包进行充分的管理，可能还具有以下字段：

Section: 这个字段申明软件的类别，常见的有`utils`、`net`、`mail`、`text`、`x11`等；

Priority: 这个字段申明软件对于系统的重要程度，如`required`、`standard`、`optional`、`extra`等；

Essential: 这个字段申明是否是系统最基本的软件包（选项为yes/no），如果是的话，这就表明该软件是维持系统稳定和正常运行的软件包，不允许任何形式的卸载（除非进行强制性的卸

载)

Architecture:申明软件包结构，如基于`i386`, `amd64`, `m68k`, `sparc`, `alpha`, `powerpc` 等；

Source: 软件包的源代码名称；

Depends: 软件所依赖的其他软件包和库文件。如果是依赖多个软件包和库文件，彼此之间采用逗号隔开；

Pre-Depends: 软件安装前必须安装、配置依赖性的软件包和库文件，它常常用于必须的预运行脚本需求；

Recommends: 这个字段表明推荐的安装的其他软件包和库文件；

Suggests: 建议安装的其他软件包和库文件。

对于control，这里有一个完整的例子：

Package: bioinfoserv-arb

Version: 2007\_14\_08

Section: BioInfoServ

Priority: optional

Depends: bioinfoserv-base-directories (>= 1.0-1), xviewg (>= 3.2p1.4), xfig (>= 1:3), libstdc++2.10-glibc2.2

Suggests: fig2ps

Architecture: i386

Installed-Size: 26104

Maintainer: Mingwei Liu <>

Provides: bioinfoserv-arb

Description: The ARB software is a graphically oriented package comprising various tools for sequence database handling and data analysis.

If you want to print your graphs you probably need to install the suggested fig2ps package.  
preinst: 这个文件是软件安装前所要进行的工作，工作执行会依据其中脚本进行；

postinst这个文件包含了软件在进行正常目录文件拷贝到系统后，所需要执行的配置工作。

prerm :软件卸载前需要执行的脚本

postrm: 软件卸载后需要执行的脚本现在来看看如何修订一个已有的deb包软件

=====

=====

debian制作DEB包(在root权限下)，打包位置随意。

#建立要打包软件文件夹，如

mkdir Cydia

cd Cydia

#依据程序的安装路径建立文件夹,并将相应程序添加到文件夹。  
如

mkdir Applications

mkdir var/mobile/Documents (游戏类需要这个目录，其他也有可能需要)

mkdir \*\*\* (要依据程序要求来添加)

#建立DEBIAN文件夹

mkdir DEBIAN

#在DEBIAN目录下创建一个control文件,并加入相关内容。

touch DEBIAN/control (也可以直接使用vi DEBIAN/control编辑保存)

#编辑control

vi DEBIAN/control

#相关内容 (注意结尾必须空一行) :

Package: soft (程序名称)

Version: 1.0.1 (版本)

Section: utils (程序类别)

Architecture: iphoneos-arm (程序格式)

Installed-Size: 512 (大小)

Maintainer: your <your\_email@gmail.com> (打包人和联系方式)

Description: soft package (程序说明)

(此处必须空一行再结束)

注: 此文件也可以先在电脑上编辑 (使用文本编辑就可以, 完成后去掉.txt),再传到打包目录里。

#在DEBIAN里还可以根据需要设置脚本文件

preinst

在Deb包文件解包之前, 将会运行该脚本。许多“preinst”脚本的

任务是停止作用于待升级软件包的服务，直到软件包安装或升级完成。

### postinst

该脚本的主要任务是完成安装包时的配置工作。许多“postinst”脚本负责执行有关命令为新安装或升级的软件重启服务。

### prerm

该脚本负责停止与软件包相关联的daemon服务。它在删除软件包关联文件之前执行。

### postrm

该脚本负责修改软件包链接或文件关联，或删除由它创建的文件。

#postinst 如：

```
#!/bin/sh
```

```
if [ "$1" = "configure" ]; then
```

```
/Applications/MobileLog.app/MobileLog -install
```

```
/bin/launchctl load -wF /System/Library/LaunchDaemons/
```

```
com.iXtension.MobileLogDaemon.plist
```

```
fi
```

#prerm 如：

```
#!/bin/sh
```

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

```
/Applications/MobileLog.app/MobileLog -uninstall
```

```
/bin/launchctl unload -wF /System/Library/LaunchDaemons/  
com.iXtension.MobileLogDaemon.plist  
fi
```

#如果DEBIAN目录中含有postinst 、 prerm等执行文件  
chmod -R 755 DEBIAN

#退出打包软件文件夹，生成DEB

```
dpkg-deb --build Cydia
```

```
=====
```

有时候安装自己打包的deb包时报如下错误：

Selecting previously deselected package initrd-deb.

(Reading database ... 71153 files and directories currently  
installed.)

Unpacking initrd-deb (from initrd-vstools\_1.0\_amd64.deb) ...

dpkg: error processing initrd-vstools\_1.0\_amd64.deb (--install):

trying to overwrite `/boot/initrd-vstools.img', which is also in  
package initrd-deb-2

dpkg-deb: subprocess paste killed by signal (Broken pipe)

Errors were encountered while processing:

initrd-vstools\_1.0\_amd64.deb

主要意思是说，已经有一个deb已经安装了相同的文件，所以默认退出安装，只要把原来安装的文件给卸载掉，再次进行安装就可以了。

下面为实践内容：

所有的目录以及文件：

mydeb

|----DEBIAN

    |-----control

        |-----postinst

    |-----postrm

|----boot

    |----- initrd-vstools.img

在任意目录下创建如上所示的目录以及文件

```
# mkdir -p /root/mydeb # 在该目录下存放生成  
deb包的文件以及目录
```

```
# mkdir -p /root/mydeb/DEBIAN #目录名必须大写
```

```
# mkdir -p /root/mydeb/boot # 将文件安装到/boot目  
录下
```

```
# touch /root/mydeb/DEBIAN/control # 必须要有该文件
```

```
# touch /root/mydeb/DEBIAN/postinst # 软件安装完后，执行该  
Shell脚本
```

```
# touch /root/mydeb/DEBIAN/postrm # 软件卸载后，执行该  
Shell脚本
```

```
# touch /root/mydeb/boot/initrd-vstools.img # 所谓的“软件”程
```

序，这里就只是一个空文件

control文件内容：

Package: my-deb     (软件名称，中间不能有空格)

Version: 1           (软件版本)

Section: utils       (软件类别)

Priority: optional    (软件对于系统的重要程度)

Architecture: amd64   (软件所支持的平台架构)

Maintainer: xxxxxx <> (打包人和联系方式)

Description: my first deb (对软件所的描述)

postinst文件内容（软件安装完后，执行该Shell脚本，一般用来配置软件执行环境，必须以“#!/bin/sh”为首行，然后给该脚本赋予可执行权限：chmod +x postinst）：

```
#!/bin/sh
```

```
echo "my deb" > /root/mydeb.log
```

postrm文件内容（软件卸载后，执行该Shell脚本，一般作为清理收尾工作，必须以“#!/bin/sh”为首行，然后给该脚本赋予可执行权限：chmod +x postrm）：

```
#!/bin/sh
```

```
rm -rf /root/mydeb.log
```

给mydeb目录打包：

```
# dpkg -b mydeb mydeb-1.deb     # 第一个参数为将要打包的目录名，
```



# 第二个参数为生成包的名称。

安装deb包：

```
# dpkg -i mydeb-1.deb    # 将initrd-vstools.img复制到/boot目  
录下后，执行postinst,  
                        # postinst脚本在/root目录下生成一  
个含有"my deb"字符的mydeb.log文件
```

卸载deb包：

```
# dpkg -r my-deb    # 这里要卸载的包名为control文件  
Package字段所定义的 my-deb 。  
                        # 将/boot目录下initrd-vstools.img删除  
后，执行posrm,  
                        # postrm脚本将/root目录下的mydeb.log  
文件删除
```

查看deb包是否安装：

```
# dpkg -s my-deb    # 这里要卸载的包名为control文件  
Package字段所定义的 my-deb
```

查看deb包文件内容：

```
# dpkg -c mydeb-1.deb
```

查看当前目录某个deb包的信息：

```
# dpkg --info mydeb-1.deb
```

解压deb包中所要安装的文件

```
# dpkg -x mydeb-1.deb mydeb-1 # 第一个参数为所要解压的deb包，这里为 mydeb-1.deb
```

```
# 第二个参数为将deb包解压到指定的目录，这里为 mydeb-1
```

解压deb包中DEBIAN目录下的文件（至少包含control文件）

```
# dpkg -e mydeb-1.deb mydeb-1/DEBIAN # 第一个参数为所要解压的deb包，
```

```
# 这里为
```

```
mydeb-1.deb
```

```
# 第二个参数为
```

```
将deb包解压到指定的目录，
```

```
# 这里为
```

```
mydeb-1/DEBIAN
```