Deb 包的制作

happyaron

(2010-10-17)

deb 格式是 Debian 系专用安装包格式，配合 APT 软件管理系统，成为了当前在 linux 软件中非常流行的一种安装包。Debian 和 ubuntu 软件安装用的都是 deb 包。

但是很多人制作 deb 包时，都是从 rpm 转为 deb，或者使用 *dpkg -b* 进行转换。虽然这两种方法可以制作出来 deb，但是显然很粗糙，今天介绍一下正规方法：使用 debhelper 制作 deb。今天用一个例子来和大家一起做个包试试看。

我们的例子是 gwrite 这个软件，软件主页： http://code.google.com/p/gwrite/。您可以点击此处下载源码。这个源码包在 Debian 项目里的术语叫做 **upstream tarball**，也就是上游发行的代码包。

源码下载完成后，我们在主目录里新建一个文件夹，例如叫 packaging，然后执行如下命令：

mkdir ~/packaging

cd ~/packaging

把刚才下载的 .tar.gz 文件放到这个文件夹里。然后用命令解压这个文件：

tar zxf gwrite-0.5.0.tar.gz

（不要使用图形化工具解压，因为会造成权限混乱），进入解压出来的目录：

cd gwrite-0.5.0

然后大家要安装几个软件包，依次介绍一下。第一个要安装的是 debhelper：

sudo aptitude install debhelper

debhelper 软件包里是各种脚本，可以帮助我们接下来的打包工作。下一个要安装的是 dh-make：

sudo aptitude install dh-make

dh-make 包提供了我们需要用到的 dh_make 命令。这个命令用于根据上游 tarball 生成我们 deb 包模板。现在我们已经 cd 到解压好的程序目录，现在的文件夹路径是 ~/packaging/gwrite-0.5.0 。然后我们执行这个命令：

dh_make -e “First Last <youremail@example.com>” -f ../gwrite-0.5.0.tar.gz

其中 First Last 是姓名，比如我是 Aron Xu。<>里是你的 email 地址。 -f 后面是上游 tarball 的路径。如果上面那个命令报错，请使用这个：

dh_make -e youremail@example.com -f ../gwrite-0.5.0.tar.gz

(../ 的意思是当前目录的上层目录）接下来程序会提示：

Type of package: single binary, indep binary, multiple binary, library, kernel module, kernel patch or cdbs?

我们输入 s，表示这个源代码包只生成一个 deb。输入 s，按回车。然后会显示一些信息，回车确认这时候再看当前的目录，会多出一个 debian/ 文件夹。上级目录里，会出现 gwrite_0.5.0.orig.tar.gz 文件。

接下来下面进到 debian/ 目录，第一个文件是 changelog。Changelog 顾名思义，是说软件版本历史的。

gwrite (0.5.0-1) unstable; urgency=low

Initial release (Closes: #nnnn) <nnnn is the bug number of your ITP>

-- Aron Xu <happyaron.xu@gmail.com> Wed, 17 Nov 2010 20:21:24 +0800

文件里面的姓名和邮箱，是用来识别这个包制作者的，当然就是在动手的各位啦。（changelog 里没有写中文的，都是英文。不知道 deb 包是否允许用中文。）changelog 是必须得是文件，没有这个文件接下来的步骤会出错。

然后是 compat 文件，里面就一个数字，现在是 7。这个数字是 debhelper 的版本。不管它就可以。

下一个文件是 control 文件：

Source: gwrite 表示源码包的名称；

Section: unknown 这行表示软件分类；

Priority: extra 代表优先级，一般可以写 optional 或者 extra。其中 optional 是普通包最常用的，extra 的优先级相比则还要再低一些；

Maintainer: Aron Xu <happyaron.xu@gmail.com>这行是软件包维护者，格式为：First Last <youremail@example.com>；

Build-Depends: debhelper (>= 7)，这行是编译依赖关系，也就是说要安装哪些软件包才可以编译这个程序。这个例子中，debhelper 是必须的。我们手里这个例子需要这些程序才能编译：Build-Depends: debhelper (>= 7), python, python-support, python-setuptools, python-distutils-extra。这些也是编译基本的 python 程序缩必须的包。（ 依赖需要自己分析 ）

Standards-Version: 3.8.3，这行是使用的 Debian Policy 版本，目前最新的是 3.9.1。所以我们写成：Standards-Version: 3.9.1；

Homepage: <insert the upstream URL, if relevant> ，这行是上游的首页地址。我们是从 http://code.google.com/p/gwrite 下载的，所以写：Homepage: http://code.google.com/p/gwrite。

下面是一个空行，表示 Source 部分结束；

Package: gwrite，这是说 deb 包的名叫 gwrite，将来 apt-get 安装的时候，就用这个名字。

Architecture: any ，这是指要编译的硬件构架。any 代表所有构架。all 代表这个软件包是跨平台的，比如说同 python 程序文件可以在所有平台上跑，我们打包的是 python 包，把 any 改成 all。如果打一个 C 程序的包，则用 any；

**Depends: ${shlibs:Depends}, ${misc:Depends}**，这一行是 deb 包的依赖关系。${shlibs:Depends}, ${misc:Depends} 是两个变了，表示 debhelper 自动检测依赖；这个例子中，我们还要添加 python, python-gtk2, python-jswebkit, python-webkit, mimetex, wv。添加完的 Depends 是这样的：Depends: ${shlibs:Depends}, ${misc:Depends}, python, python-gtk2, python-jswebkit, python-webkit, mimetex, wv；

Kandu: dpkg-dev，会检测自己的 host，把那个项在编译时替换成当前的。所以写 any 即可；

Description: <insert up to 60 chars description>，这行是简单的描述，要在 60 个字符以内，要用英语。从语言学来讲，这应该是一个名词性的成分，比如手里这个就写成： simple GTK+

HTML5 rich text editor。写完的样子是：

Description: simple GTK+ HTML5 rich text editor。

<insert long description, indented with spaces> ，这里写的是长描述。也是用英语，应该是完整的句子，每行大概写 70 个字符，不要太多，没写完可以换行，每行的开头都要有一个空格，示例：*gWrite is a simple GTK+ HTML5 WYSIWYG editor, focusing on writing and simple text formating. It can automatically generate a table of contents based on the document structure.* 每行开头都有一个半角空格。这样我们 control 文件就写完了。

Control 文档示例：

```
Source: gwrite
Section: editors
Priority: optional
Maintainer: Aron Xu <happyaron.xu@gmail.com>
Build-Depends: cdbs, debhelper (>= 7), python, python-support, python-setuptools, python-distutils-extra
Standards-Version: 3.9.1
Homepage: http://code.google.com/p/gwrite

Package: gwrite
Architecture: all
Depends: ${shlibs:Depends}, ${misc:Depends}, python, python-gtk2, python-jswebkit, python-webkit, mimetex, wv
Description: simple GTK+ HTML5 rich text editor
 gWrite is a simple GTK+ HTML5 WYSIWYG editor, focusing on writing
 and simple text formating. It can automatically generate a table of
 contents based on the document structure.
 .
 It aims to be lighter than OOWrite & OOWeb, and to be as useful as them.
```

下面是 copyright 文件 。这个文件里写的是版权内容。版权的写法比较复杂，今天先按下不表。（您可以参考范本进行修改）

再看文件夹里有很多 .ex 文件。这些文件都是某些功能性的脚本，通常来说，大多数包都只需要一个 watch.ex。图形化程序还需要 menu.ex。watch.ex 是 debian 监视上游新版本用的工具，不进入官方仓库的包用不上，今天不讲写法。menu.ex 是桌面菜单项，我们要使用它，所以把他重命名为 menu。也就是去掉.ex 后缀。默认模板：

```
?package(gwrite):needs="X11|text|vc|wm" section="Applications/see-menu-manual"\ title="gwrite" command="/usr/bin/gwrite"
```

needs="X11|text|vc|wm" 这句是说这个包需要什么显示平台来运行，如果是文本的程序，可以写 text，图形化程序则写 X11。**section="Applications/see-menu-manual"** 这是说程序的分类。所有可以用的选项在这里找到 http://www.debian.org/doc/packaging-manuals/menu-

policy/ch2.html#s2.1 因为例子里的程序是个编辑器，所以写成:

```
?package(gwrite):needs="X11" section="Applications/Editors"\
title="gwrite" command="/usr/bin/gwrite"
```

其余的 .ex 文件和.EX 文件在例子里都用不上，删除。

README.Debian README.source 的用处是，如果你想就这个包做点什么说明，那么写在这里 README.Debian 是写给最终用户的。README.source 是关于源代码包有什么问题，如果没有什么要在里面说的，删掉。docs 文件是 dh_make 自动检测的上游文档列表，如果是空的，删掉。我们的例子里它就是空的，所以删了。现在要在文件夹里创建一个名为 pycompat 的文件，里面写 2 就行了，这个文件没有什么变的，就是 python 包就加上它，不是 python 包不需要它，pycompat 文件里只有一个数字：2 。

在当前目录下创建一个文件夹: source，然后在 source 目录下创建一个名为 format 的文本文件，里面写一行文字：3.0 (quilt)。这代表使用 3.0 (quilt) 的源代码格式。

最后剩下 rules 文件了。这个文件是编译 deb 包的主控文件，它本身是一个 Makefile，但是不要往里乱写 target。90%的情况下，默认的 rules 就可以满足我们的要求，现在 debian/ 目录下的工作基本完成了。

我们开始尝试编译这个包。

编译的第一步，是安装所有编译依赖，也就是写在 control 文件的 Build-Depends 项里的内容：debhelper python-support python-setuptool python-distutils-extra。当前的例子里是这些：

sudo aptitude install debhelper python-support python-setuptool python-distutils-extra

安装完之后，先检查自己是否在源代码目录里。例子里也就是 gwrite-0.5.0 目录。在 gwrite-0.5.0 里运行：

dpkg-buildpackage -rfakeroot（不需要 root，直接运行即可）

如果执行成功，在父目录里就会出现 .deb .build 和 .changes 文件；如果出现错误，就要根据提示再从头找原因了。一个最简单 deb 包的制作例程，就是这样。完整地展示给大家了。

参考材料：http://www.debian.org/doc/maint-guide/index.zh-cn.html（已重新翻译最新版）
示例内容及下载：http://ftp.de.debian.org/debian/pool/main/g/gwrite/gwrite_0.5.0-1.debian.tar.gz（解包即可）

from：http://logs.ubuntu-eu.org/free/2010/11/17/%23ubuntu-cn.html