

Taller de creación de paquetes para Debian



Ana Beatriz Guerrero López <ana@debian.org>
Dudesconf, 13 Abril, Coruña

Resumen del proceso de empaquetado

- 1) Obtener el código fuente original.
 - Comprobar que la licencia compatible con las DSFG.
- 2) Anunciar “Intend to package” (ITP) o renombrar RFP (Request For Package)
 - Comprobar si existen ITPs previos o RFP
- 3) Añadir cambios específicos para Debian
- 4) Construir paquete fuente y binarios para Debian
- 5) Comprobar y arreglar problemas (y repetir)
- 6) Desde aquí, seguir con las actualizaciones

Resumen del proceso de empaquetado (alternativo)

- 1) En lugar de empaquetar nuevo software:
 - Adoptar un paquete huérfano (O)
 - Adoptar un paquete que busca nuevo mantenedor/a (RFA)
 - Ayudar en un paquete que necesita ayuda (RFH)
- 2) Añadir cambios específicos para adoptar paquete. Si es nueva version los requeridos para actualizar.
- 3) Construir paquete fuente y binarios para Debian
- 4) Comprobar y arreglar problemas (y repetir)
- 5) Desde aquí, seguir con las actualizaciones

Comprobar licencia

- Comprobar que **todos** los archivos del programa tienen una licencia que cumple las DFSG.
- Y anotar **todas** las licencias que tienen los archivos de la aplicación que estás distribuyendo. Tener cuidado especial con imágenes, sonidos,...
- Algunas licencias que son compatibles con las DFSG son: Artistic, BSD, GPL, GPL-2, LGPL, LGPL-2, LGPL-2.1

Como rellenar una ITP

- Una ITP es un bug contra el pseudopackage wnpp, tiene la siguiente estructura:

```
Subject: ITP: PACKAGENAME -- A DESCRIPTION
Package: wnpp
Version: N/A; reported 2002-01-30
Severity: wishlist
```

```
* Package name : PACKAGENAME
Version : x.y.z
Upstream Author : Name <somebody@some.org>
* URL : http://www.some.org/
* License : (GPL, LGPL, BSD, MIT/X, etc.)
Description : A DESCRIPTION
```

- Se puede enviar un mail directamente a submit@bugs.debian.org con este contenido, o usar reportbug (urg!)
- Resto de casos (O, RFP) <http://www.debian.org/devel/wnpp/>

Ah, antes que nada...

- Paquetes necesarios siempre que vamos a empaquetar:

```
apt-get install build-essential devscripts dh-make fakeroot
```

- Variables de entorno:

```
export DEBEMAIL=tu@email.com  
export DEBFULLNAME='Tu nombre'
```

Código original (upstream)

- Normalmente un .tar.gz de una web
 - A veces en formato diferente (tar.bz2) o sólo en un sistema de control de versiones (CVS/SVN/...), en cuyo caso es necesario crear el .tar.gz
- El nombre debe ser `foo_1.2.3.orig.tar.gz`
 - A ser posible **sólo** renombrar el original
- El directorio con el código una vez descomprimido debe tener la forma: `foo-1.2.3`

dh-make

Podemos usar dh-make de la siguiente forma:

```
tar xvfz tintin-1.96.9.tar.gz
mv tt tintin++-1.96.9
cd tintin++-1.96.9/
dh_make -f ../tintin-1.96.9.tar.gz
Type of package: single binary, multiple binary, library,
kernel module or cdfs?
[s/m/l/k/b] s
```

```
Maintainer name : Ana Beatriz Guerrero Lopez
Email-Address   : ana@debian.org
Date            : Fri, 13 Apr 2007 11:51:00 +0200
Package Name    : tintin++
Version         : 1.96.9
License         : blank
Type of Package : Single
Hit <enter> to confirm:
```


Ficheros bajo debian/ - la parte fácil

Algunos ficheros no los necesitaremos, podemos borrar:

- `cron.d.ex` (ejemplo para un paquete que tenga que añadir una tarea al cron)
- `emacs-*` (archivos relacionados con emacs, borrarlos rápido que emacs es malvado ;-))
- `init.d.ex` (ejemplo para un paquete con script de inicio)
- `tintin++-default.ex` (ejemplo de un archivo de defaults `/etc/default/*`, está relacionado con los scripts de inicio)
- `tintin++-doc-base.EX` (para añadir la documentación a la base de datos general de documentos, no lo usaremos)

Ficheros bajo debian/ - la parte fácil

- `postinst.ex`, `prerm.ex`, `preinst.ex` y `postrm.ex` son los *maintainer scripts* que se ejecutan cuando el paquete se instala o se desinstala. Nuestro paquete es muy simple y no necesita nada de esto (y así muchísimos paquetes).
- `README.Debian` (para añadir información específica sobre el software en Debian).
- `manpage*` Todo binario en Debian **debe** tener una manpage que se llame como el binario. En nuestro caso, usaremos la página que incluye actualmente el paquete (llamada `tt++.6`)
- `watch.ex` (Es una buena idea crear ficheros `watch`, pero no es imprescindible y lo omitiremos).

Que ficheros nos quedan

Tras borrar los ficheros anteriores nos deberian de quedar bajo debian/

- changelog
- compat
- control
- copyright
- dirs
- docs
- menu.ex
- rules

debian/changelog

- Log de cambios en el empaquetado para Debian
 - Los cambios que se realicen relacionados con el empaquetado deben ser resumidos brevemente.
 - Se usa también para el cierre de bugs.

```
tintin++ (1.96.9-1) unstable; urgency=low
```

```
* Initial release (Closes: #nnnn)
```

```
* Improved manpage.
```

```
-- Ana Beatriz Guerrero Lopez <ana@debian.org> Fri, 13 Apr 2007 11:51:00 +0100
```

debian/compat

Es un solo número con la versión de compatibilidad de debhelper que usaremos.

Actualmente es la version 5, así que no hay que tocar nada :-)

debian/control

- **Metadatos de los paquetes fuente y binarios**

Source: tintin++

Section: games

Priority: optional

Maintainer: Ana Beatriz Guerrero Lopez <ana@debian.org>

Build-Depends: debhelper (>= 5.0.0), libreadline5-dev,
libncurses5-dev, zlib1g-dev

Standards-Version: 3.7.2

debian/control

Package: tintin++

Architecture: any

Section: games

Priority: optional

Depends: \${shlibs:Depends}

Description: classic text-based MUD client

Tintin++ is telnet client specialized to play MUDs (Multi-User Dungeons). It has scripting support, tab-completion, internal chat, and the advantage of the GNU readline library.

.

You can find a complete set of commands and features in the manual in /usr/share/doc/tintin++.

.

Homepage: <http://tintin.sourceforge.net/>

debian/copyright

- Copyright e información sobre la licencia
 - ¡Es esencial hacerlo correctamente! ¡Muy importante!
 - Incluir el texto completo (excepto para las licencias GPL, LGPL y otras que estén en `/usr/share/common-licenses`)
- Texto más o menos libre.
- También: dónde descargar el fuente original, quién realizó el empaquetado, etc.
- Licencia del empaquetado.

`debian/dirs`

Especifica los directorios que se necesitan pero que por alguna razón no se crean en el proceso de instalación normal.

En este caso, al ser tintin un “juego”, el ejecutable debe ir en `usr/games` en lugar de `usr/bin`, y el contenido del fichero será:

`usr/games`

debian/docs

Contiene el listado de ficheros de configuración
que se instalarán bajo

/usr/share/doc/<paquete>/

En este caso:

BUGS

CREDITS

FAQ

README

TODO

docs/tintin19.txt

debian/menu

Es para tener una entrada en el menu Debian.

```
?package (tintin++) :needs="text" \  
section="Games/Adventure" \  
title="Tintin++"  command="/usr/games/tt++"
```

debian/rules

- Reglas para compilar y construir el paquete
 - Técnicamente un Makefile con reglas específicas: build, binary, binary-arch, binary-indep, clean
 - La mayoría de las veces utilizando asistentes. (debhelper)
- Algunas veces muy simple, otras muy, muy, complejo.
 - La integración en el sistema puede resultar difícil

debian/rules: resumen

- `debian/rules build`: **compila el programa**
- `debian/rules binary`: **crea el paquete binario**
 - Instala el programa bajo `debian/tmp`
 - Pone los metadatos de empaquetado en `debian/tmp/DEBIAN`
- `debian/rules clean`: **limpieza**

debian/rules

Muy largo para ponerlo aquí.

```
apt-get source tintin++  
cd tintin*debian/  
cat rules
```

Creación del paquete

- Ejecuta `debian/rules` a mano hasta que todos los ficheros de empaquetado sean correctos
- `fakeroot dpkg-buildpackage -us -uc`
(o `debuild`)
 - Ejecuta el proceso completo de creación del paquete (sin firmar `*.dsc` y `*.changes`)
 - También crea un paquete fuente
- `sudo pbuilder build foo_1.0-1.dsc`
 - Ejecuta el proceso completo en un "chroot" para comprobar dependencias (sin firmar)
 - Puede esperar a después (avanzado)

Resultado

- `foo_1.0.orig.tar.gz`
- `foo_1.0-1.diff.gz`
- `foo_1.0-1.dsc`
- `foo_1.0-1_i386.changes`
- `foo_1.0-1_i386.deb`

Comprobando el paquete

- `lintian -i *.changes`
`linda -i *.changes`
 - Comprobación automática de problemas comunes
- Pruebas de instalación:
 - Instalar en un sistema nuevo, eliminar
 - Actualizar desde versión anterior
 - Actualizar a una versión anterior, si es posible
 - Posiblemente usar un chroot (ver debootstrap)
 - `piuparts`

¿Y después?

- Ponerlo en una web
 - Posiblemente en tu propio repositorio apt (para que `apt-get install` funcione), necesita un poco de configuración `apt-ftparchive`
- Firmar con GPG los ficheros `.changes` y `.dsc` y subir a Debian
 - Sólo desarrolladores oficiales, otros a través de sponsors
 - Los paquetes “NEW” se comprueban manualmente antes de entrar a Debian

¿Por dónde seguir?

- <http://www.debian.org> para aprender más
 - Developers' Corner (<http://www.debian.org/devel>)
 - New Maintainers' guide (<http://www.debian.org/doc/maint-guide>)
 - Debian Policy Manual (<http://www.debian.org/doc/debian-policy>)
 - Bug Tracking System, BTS (<http://bugs.debian.org>)
 - Work-Needing and Prospective Packages
<http://www.debian.org/devel/wnpp/>
- Seguir discusiones del proyecto
 - <http://lists.debian.org>
 - irc.debian.org / irc.oftc.net

¡Muchas gracias dudes! =)



Esquema del proceso de empaquetado

Código original (upstream)

Ficheros de empaquetado Debian

