

Empaquetar software para Debian

Una breve introduccion

Maximiliano Curia

18 de Agosto del 2008

- 1 Introducción
- 2 Para mí
- 3 Para los demás
- 4 Para Debian
- 5 Más allá
- 6 Documentación

- ¿Qué es empaquetar?
- ¿Por qué me puede interesar?
- ¿De qué se trata?

Me decidí

- Revisar en wnpp
- Revisar licencias
- Revisar documentación

- Compilarlo y probarlo
 - ./configure
 - make
 - make install

Archivos obligatorios

- debian/control
- debian/changelog
- debian/copyright
- debian/rules y debian/compat

```
Source: vacuum
Section: games
Priority: optional
Maintainer: Maximiliano Curia <maxy@debian.org>
Build-Depends: debhelper (>= 7), libsdl-perl
Standards-Version: 3.8.0
```

```
Package: vacuum
Architecture: any
Depends: ${shlibs:Depends}, ${misc:Depends}
Description: Vacuum Magic is a fast-paced action game
  The point of the game is using your magical vacuum field to
  collect food and defend against monsters. Food and certain
  monsters can also be spat out and used as a projectile against
  other monsters.
.
Vacuum Magic can be played by up to six players, either
cooperatively, or against each other.
```

This package was debianized by Maximiliano Curia <maxy@debian.org>
on Fri, 15 Aug 2008 12:17:47 -0300.

It was downloaded from: <http://apocalypse.rulez.org/vacuum/Download>
Upstream Author(s):

UPi <upi at sourceforge.net>

Music: See the music files for individual credits

Copyright:

Copyright (C) 2008 by UPi <upi at sourceforge.net>

License:

This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
version 2, as published by the Free Software Foundation.

The Debian packaging is (C) 2008, Maximiliano Curia
<maxy@debian.org> and is licensed under the GPL, see
'/usr/share/common-licenses/GPL'.


```
vacuum (0.4-1) unstable; urgency=low
```

```
 * Initial release. (Closes: #500000)
```

```
-- Maximiliano Curia <maxy@debian.org> Fri, 15 Aug 2008 12:01:45 -0300
```

debian/rules (debhelper)

```
#!/usr/bin/make -f

# Uncomment this to turn on verbose mode.
#export DH_VERBOSE=1

# These are used for cross-compiling and for saving the configure script
# from having to guess our platform (since we know it already)
DEB_HOST_GNU_TYPE    ?= $(shell dpkg-architecture -qDEB_HOST_GNU_TYPE)
DEB_BUILD_GNU_TYPE   ?= $(shell dpkg-architecture -qDEB_BUILD_GNU_TYPE)
ifneq ($(DEB_HOST_GNU_TYPE),$(DEB_BUILD_GNU_TYPE))
CROSS= --build $(DEB_BUILD_GNU_TYPE) --host $(DEB_HOST_GNU_TYPE)
else
CROSS= --build $(DEB_BUILD_GNU_TYPE)
endif

config.status: configure
    dh_testdir
    # Add here commands to configure the package.
ifneq "$(wildcard /usr/share/misc/config.sub)" ""
    cp -f /usr/share/misc/config.sub config.sub
endif
ifneq "$(wildcard /usr/share/misc/config.guess)" ""
    cp -f /usr/share/misc/config.guess config.guess
endif
    ./configure $(CROSS) --prefix=/usr --mandir=\${prefix}/share/man
    --infodir=\${prefix}/share/info CFLAGS="\$(CFLAGS)"
    LDFLAGS="-Wl,-z,defs"

build: build-stamp
```

debian/rules (debhelper cont..)

```
build-stamp: config.status
    dh_testdir
    # Add here commands to compile the package.
    $(MAKE)
    #docbook-to-man debian/vacuum.sgml > vacuum.1
    touch $@

clean:
    dh_testdir
    dh_testroot
    rm -f build-stamp
    # Add here commands to clean up after the build process.
    [ ! -f Makefile ] || $(MAKE) distclean
    rm -f config.sub config.guess
    dh_clean

install: build
    dh_testdir
    dh_testroot
    dh_clean -k
    dh_installdirs
    # Add here commands to install the package into debian/vacuum.
    $(MAKE) DESTDIR=$(CURDIR)/debian/vacuum install

# Build architecture-independent files here.
binary-indep: build install
# We have nothing to do by default.
```

debian/rules (debhelper cont....)

```
# Build architecture-dependent files here.
binary-arch: build install
    dh_testdir
    dh_testroot
    dh_installchangelogs ChangeLog
    dh_installdocs
    dh_installexamples
#
# dh_install
# dh_installmenu
# dh_installdebconf
# dh_installogrotate
# dh_installemacsen
# dh_installpam
# dh_installmime
# dh_python
# dh_installinit
# dh_installdocs
# dh_installinfo
dh_installman
dh_link
dh_strip
dh_compress
dh_fixperms
# dh_perl
# dh_makeshlibs
dh_installdeb
dh_shlibdeps
dh_gencontrol
dh_md5sums
dh_builddeb
```

debian/rules (debhelper cont.....)

```
binary: binary-indep binary-arch  
.PHONY: build clean binary-indep binary-arch binary install
```

debian/rules

```
#!/usr/bin/make -f
```

```
%:
```

```
dh $@
```

debian/compat

7

Usando dh un poco más detallista

debian/rules

```
#!/usr/bin/make -f
```

```
build:
```

```
dh build --before dh_auto_configure  
dh_auto_configure -- '--bindir=/usr/games'  
dh build --after dh_auto_configure
```

```
%:
```

```
dh $@
```

Generar el paquete

- `# dh_make`
- `apt-get install build-essential fakeroot debhelper`
- `dpkg-buildpackage -rfakeroot`

Cosas a tener en cuenta

- Dependencias (pbuilder)
- debian/dirs
- debian/docs

Cosas a tener en cuenta

- Crear un simple archivo de paquetes (apt-ftarchive)
- Archivos de configuración (debian/conffiles)
- Código fuente inalterado, uso de parches (dpatch, quilt o nuevo dpkg)
- menu, manpage, etc

Cosas a tener en cuenta

- RFP, ITP, contactar al autor
- Llave para firmar el paquete
- Conseguir un mentor/esponsor

Más cosas para tener en cuenta

- Archivos debian/watch
- debconf
- Múltiples paquetes del mismo código fuente
- Bibliotecas y símbolos con versiones

- Debian Policy (debian-policy)
- Guía para el mantenedor (maint-guide)
- Referencia para el Debian Developer (developers-reference)
- Políticas particulares para software particulares (perl, python, etc)
- La documentación y el código fuente