

# Nhopkg Universal Packaging Standards

(Estándares de Empaquetado Universal con Nhopkg)

Versión en pruebas, 28 de marzo de 2009

Copyright (C) 2009 Jaime Gil de Sagredo <[jaimegildesagredo@gmail.com](mailto:jaimegildesagredo@gmail.com)>

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

## Preámbulo

**Nhopkg** pretende crear un repositorio universal de software libre, para poder ser instalado en cualquier sistema de tipo Unix sin necesidad de conocimientos avanzados en administración de estos sistemas. Actualmente ya existen otros sistemas de gestión de paquetes con sus respectivos front-ends para la administración de dependencias e instalación desde repositorios de software, como pueden ser rpm (yum), dpkg (apt-get) y otros, como **Nhopkg**, que integran el uso de repositorios y la resolución de dependencias desde los mismos, en el mismo gestor de paquetes, como pueda ser Pacman (Arch Linux). A pesar de todo ésto, ninguno de los mencionados anteriormente es capaz de instalar, este software que ofrecen, en otras distribuciones GNU/Linux distintas, o, en caso de poder, esta instalación no es lo más adecuada posible debido a diversas circunstancias, como el uso de distintos nombres para las dependencias, la mala compatibilidad de esos paquetes con otros sistemas, etc. Por ello, en este texto se pretende llegar a unos estándares a partir de los cuales se puedan crear paquetes de software, con **Nhopkg**, los cuales se puedan instalar y funcionar correctamente en cualquier sistema, ya sea GNU/Linux o cualquier otro derivado de Unix (véase BSD especialmente).

Este texto consta de una serie de apartados en los que se va guiando al usuario o administrador de un sistema en el camino de crear los paquetes de la forma más compatible posible.

Se ha usado como guía el tipo de exposición que se hace en la Licencia Pública General GNU (GPL).

A partir de aquí se expondrán ordenadamente los estándares para la consecución de paquetes de software universales a partir de **Nhopkg**.

# ESTÁNDARES DE EMPAQUETADO UNIVERSAL

## 1. Software oficial de los repositorios.

Todo el software que se vaya a encontrar en los repositorios oficiales de Nhopkg será única y exclusivamente Software Libre. Se pretende buscar una forma competitiva de suministrar software de primera calidad y 100% libre, quedando fuera de este repositorio cualquier programa o librería que incumpla esta primera prioridad indispensable para encontrarse en los repositorios oficiales de Nhopkg.

Este software deberá estar licenciado preferentemente con licencias GPL (sin importar la versión de éstas) o con cualquier otra licencia libre que garantice los fundamentos del Software Libre.

Para cumplir dichas licencias, será completamente necesario la suministración del código fuente, en formato .src.nho, a través de los repositorios.

También quedará excluido del repositorio cualquier software, que aun siendo libre, dependa directamente de otro que no lo sea.

## 2. Datos básicos del paquete.

A la hora de crear un paquete de fuentes, el asistente de Nhopkg realiza una serie de preguntas, como el nombre del usuario, su email y varios datos sobre el paquete. Es muy importante rellenar estos datos de forma correcta, ya que supondrán uno de los puntos básicos a la hora de que un paquete sea universal.

Los datos a destacar son el nombre del paquete, que deberá, únicamente, contener el nombre del software, la versión deberá ser una serie de números separados por puntos. La versión la proporciona el desarrollador de cada software, y no deberá ser modificada. Por último, la revisión de cada software se empezará a contar por cada mantenedor de software, y empezará desde 1 cada vez que se cambie de versión.

Otros datos importantes son el grupo al que pertenece el software, la licencia que usa (como se ve más arriba, deberá ser libre), la página web del proyecto y la descripción, básica, del programa.

## 3. Parches incluidos en el paquete.

Como se especificó en el punto primero, los únicos parches que se deberán incluir en un paquete serán aquellos suministrados por el propio autor o los presentes en los libros de Linux From Scratch.

## 4. Especificación de dependencias.

Nhopkg permite la inclusión de dependencias a través del asistente de creación de paquetes fuente. Esta inclusión puede hacerse de dos maneras. Por un lado se puede suministrar el nombre de un paquete instalado con Nhopkg o que se encuentre en repositorios, del cual se extraerán los archivos que lo forman, para completar las dependencias, o bien se puede indicar el nombre completo de un archivo, el cual deberá estar en el sistema del creador del paquete. Como ejemplo, si un paquete dependiese de Glibc, podríamos seleccionar como dependencia glibc o /lib/libc.so.6.

Aunque los dos métodos son válidos y correctos, desde aquí se quiere hacer hincapié en el uso de la segunda opción, por ser más correcta.

## 5. Modo de compilación.

En los repositorios de Nhopkg se pretende suministrar software vanilla. Es decir, el software que se empaquete deberá ser tal cual lo liberaron sus desarrolladores, tan solo pudiendo contener parches

de seguridad y resolución de bugs. Estos parches únicamente serán aceptados si son provistos por el propio desarrollador del software o bien si están suministrado por el proyecto Linux From Scratch.

Los comandos que se utilicen para la compilación del software serán expresamente obtenidos de los proporcionados por los proyectos LFS, BLFS y (B)CLFS. Si no se dispusiera de los pasos de compilación de dicho software en las páginas mencionadas anteriormente, se procedería, a criterio del creador del paquete, a escoger los comandos de compilación idóneos para la instalación de dicho software, siempre siguiendo las pautas expuestas anteriormente (software vanilla, lo que implica que se compile el software tal cual fue liberado por el desarrollador).

## 6. Comandos para configurar la instalación y la desinstalación del software.

Nhopkg posibilita la opción de añadir comandos que se ejecutarán después de la instalación y de la eliminación de un paquete. Los comandos de post instalación deberán usarse para crear archivos que no deben ir en un paquete binario, aunque también se crearán al instalar un paquete binario, como archivos de configuración, enlaces simbólicos, compresión de páginas man, etc.

Por otro lado, los comandos de post desinstalación, se deberán usar para eliminar los archivos que crearon los anteriores.

## 7. Ejemplo de empaquetado.

Ahora trataremos un ejemplo básico de empaquetado de un software. Ésto no es más que un ejemplo, por lo que siempre podría haber variaciones.

Vamos a proceder a empaquetar gimp-2.4.7. Para ello, lo primero que debemos hacer es descargarlo desde la página oficial del proyecto (<http://gimp.org>).

Desde la terminal escribimos el siguiente comando para empezar con el asistente de creación:

```
nhopkg -c Descargas/gimp-2.4.7.tar.bz2
```

Lo primero que nos preguntará el asistente serán nuestros datos personales, los cuales deberemos introducir correctamente.

*Enter your username:*

Jaime Gil

*Enter your e-mail:*

jaimegildesagredo at gmail dot com

*Enter package name (Ex: gimp):*

gimp

*Enter package version (Ex: 2.6.3):*

2.4.7

*Enter package release (Ex: 1):*

1

*Enter package license (Ex: GPLv3):*

GPLv2

*Enter package group (Ex: Graphics):*

Graphics

*Enter package arch (you can enter more than one arch separated by space. Ex: i686):*  
i686 x86\_64 ppc

*Enter package web page (Ex: <http://gimp.org/>):*  
<http://gimp.org/>

*Enter package description (Ex: GIMP is the GNU Image Manipulation Program.):*  
GIMP is the GNU Image Manipulation Program.

Ya hemos introducido los datos básicos para la creación del paquete. En estos momentos, Nhopkg re-empaquetará el tarball de gimp antes de seguir. Una vez empaquetado de nuevo, deberemos seguir contestando otras preguntas, como si le incluiremos algún parche, las dependencias, comandos de compilación, instalación, etc.

*Enter path to patch:*  
(pulamos *enter* si no queremos parches)

Ahora llega el turno de las dependencias. Necesitaremos *Gtk2*, *libart\_lgpl*, *XML::Parser*, *LibTIFF* y *libjpeg* como dependencias requeridas para compilación. Necesita de otras más, pero éstas son satisfechas si existen los paquetes anteriores, por lo que no es necesario ponerlas. Para cada una de las dependencias añadiremos dos archivos, para que sea más fiable a la hora de resolver las dependencias.

*Enter build dependency:*  
/usr/lib/libgtk-x11-2.0.so

*Enter build dependency (if necessary):*  
/usr/include/gtk-2.0/gtk/gtkrc.h

*Enter build dependency (if necessary):*  
/usr/lib/libart\_lgpl\_2.so

*Enter build dependency (if necessary):*  
/usr/include/libart-2.0/libart\_lgpl/art\_svp.h

*Enter build dependency (if necessary):*  
/usr/lib/perl5/XML/Parser.pm

*Enter build dependency (if necessary):*  
/usr/include/tiff.h

*Enter build dependency (if necessary):*  
/usr/lib/libtiff.so

*Enter build dependency (if necessary):*  
/usr/include/jpeglib.h

*Enter build dependency (if necessary):*  
/usr/lib/libjpeg.so

Una vez introducimos todas las dependencias necesarias para la compilación, ahora añadiremos, si

son necesarias, las dependencias opcionales de compilación. Nosotros vamos a añadir *librsvg* y *little cms* como dependencias opcionales.

*Enter optional build dependency:*  
/usr/lib/liblcms.so.1

Enter optional build dependency (if necessary):  
/usr/lib/librsvg-2.so

Ahora es el momento de añadir las dependencias necesarias para que Gimp arranque una vez instalado. Muchas veces estas dependencias coinciden con las de compilación. Nosotros añadiremos las mismas dependencias del primer paso, además de *glibc*.

*Enter post installation dependency:*  
/usr/lib/libgtk-x11-2.0.so

*Enter post installation dependency (if necessary):*  
/usr/lib/libart\_lgpl\_2.so

*Enter post installation dependency (if necessary):*  
/usr/lib/perl5/XML/Parser.pm

*Enter post installation dependency (if necessary):*  
/usr/lib/libtiff.so

*Enter post installation dependency (if necessary):*  
/usr/lib/libjpeg.so

*Enter post installation dependency (if necessary):*  
/lib/libc.so.6

Finalmente, como dependencias opcionales para después de la instalación, nosotros no añadiremos ninguna, aunque podrían añadirse las dependencias opcionales del primer paso.

Llegamos al paso de introducción de comandos de compilación. Se trata del paso más importante de todos. Dependiendo de qué comandos introduzcamos y si son correctos, nuestro paquete tendrá o no éxito. Para obtener los comandos de compilación podemos dirigirnos a <http://cblfs.cross-lfs.org/index.php/GIMP> o <http://www.linuxfromscratch.org/blfs/view/svn/xsoft/gimp.html>.

*Enter compilation commands:*  
./configure --prefix=/usr --sysconfdir=/etc --datadir=/usr/share  
make

*Enter instalation commands:*  
make install  
install -v -m755 -d /usr/share/doc/gimp-2.4.7  
install -v -m644 docs/{Wilber\*,keybindings.txt,quick\_reference.ps} /usr/share/doc/gimp-  
2.4.7

Después de estos dos pasos, ahora nos disponemos a introducir los comandos de configuración, tanto para después de la instalación como para después de eliminar el paquete del sistema.

*Enter post instalation commands:*

ldconfig

*Enter post remove commands:*

(presionamos *enter* si no necesitamos introducir nada)

Por último el nuevo paquete se terminará de comprimir y tendremos listo el paquete de fuentes para Nhopkg.

-- *New package is: /home/jotahacker/Descargas/gimp-2.4.7-1.src.nho*

## **8. Final y conclusiones.**

Siguiendo estos pasos podremos crear paquetes de software completamente funcionales en cualquier sistema GNU/Linux. Aunque se ha intentado llegar a la mejor opción para que sean compatibles, esta guía está aun en un proceso de desarrollo, por lo que es posible que cambien cosas con el paso del tiempo.

En este momento solo queda seguir trabajando por un estándar de empaquetado en sistemas GNU/Linux, y recordar, que no solo depende de un software, en este caso *Nhopkg*, sino de un método de empaquetado estándar, como el que intentamos conseguir con esta guía.