

Guía de procedimientos para gestión de repositorios, actualizaciones y nuevas versiones del sistema operativo Canaima GNU/Linux

(C) 2010 José Miguel Parrella Romero <bureado@debian.org>
Este documento se distribuye bajo los términos de la Licencia Pública General GNU

Última revisión: 18 de enero de 2010 1200 VET

Alcance

La presente guía cubre los procedimientos a seguir por parte de los/as desarrolladores/as, administradores/as de repositorios y líderes de versión del sistema operativo en Canaima GNU/Linux¹ para gestionar la vida de los paquetes de software *reales*: fuente y binarios, en los repositorios y medios de instalación del sistema operativo Canaima GNU/Linux.

Está fuera del alcance de este material describir la anatomía de un paquete fuente o binario del sistema APT, así como la instalación y uso avanzado de herramientas asociadas como `pbuilder` y `reprepro`²; en estos casos se remite a el/la lector/a a fuentes externas.

Generalidades

Canaima GNU/Linux utiliza el sistema de paquetes APT, difundido en cientos de sistemas operativos a nivel internacional incluyendo Debian GNU/Linux y Ubuntu Linux. En APT existen paquetes reales y virtuales. Los paquetes reales son tangibles: existen archivos físicos que se relacionan unívocamente con el paquete. Los paquetes virtuales se crean al nombrarlos en el sistema de paquetes, y están fuera del alcance de este documento.

Así mismo, en APT existen paquetes fuente, que constan del código fuente del software, tal y como sus desarrolladores/as originales lo distribuyen al público³, así como posibles cambios introducidos por los distribuidores del software, e instrucciones para construir ese código fuente. También existen paquetes binarios, resultado de la construcción de paquetes fuente.

Por requerimiento de la mayoría de las licencias bajo las cuales se distribuye el software de Canaima GNU/Linux así como por filosofía del proyecto y para facilitar el mantenimiento y difusión de nuevas arquitecturas de computadores, se tiene como **premisa** que *el público debe tener acceso a los paquetes fuente y a los paquetes binarios*.

Finalmente, es importante resaltar que la mayoría del software en Canaima GNU/Linux se toma

1 canaima.softwarelibre.gob.ve

2 Tenga en cuenta que los repositorios y distribución del software en Canaima no explotan toda la complejidad del sistema de paquetes APT ni del sistema de repositorios de Debian

3 Con la excepción de aquellos contenidos no distribuibles como software libre, como por ejemplo: fuentes tipográficas patentadas o de distribución restringida, material multimedia o porciones de código propietario.

de la versión estable de Debian GNU/Linux. Otros paquetes se incluyen desde otras distribuciones, son modificados por el equipo de Canaima GNU/Linux, se crean como metapaquetes, capas o como paquetes de software completamente nuevos.

Preparación del ambiente de pbuilder

Se utilizará preferentemente `pbuilder` como herramienta para asistir a los/as desarrolladores/as en la construcción de paquetes binarios a partir del paquete fuente. Ya que se parte de la premisa de que se va a trabajar con paquetes fuente, es esencial conocer como construir los paquetes binarios a partir de paquetes fuente.

`Pbuilder` mantiene uno o más archivos `tar` comprimidos con `gzip` que contienen un sistema base de la distribución objeto deseada. Antes de empezar a utilizarlo, hay que instalar `pbuilder` y sus dependencias y crear ese ambiente de trabajo. `Pbuilder` se ejecuta como superusuario ya que requiere acceso a dispositivos especiales y montaje de `devfs` y `procfs`.

1. `aptitude install pbuilder`
2. `pbuilder -create -distribution stable`

Se utiliza la distribución `stable` (Lenny, 5.0, a la fecha de esta publicación) ya que Canaima utiliza paquetes de `stable` en la mayoría de sus paquetes binarios⁴. Así mismo, se requiere mantener actualizado el `base.tgz` con `pbuilder -update`.

Construcción de paquetes binarios a partir del paquete fuente

En muchas ocasiones puede ser necesario construir uno o más paquetes binarios a partir del paquete fuente, por ejemplo cuando se modifica software de Debian, se incluye software de otras distribuciones, se crean capas o como aprendizaje o base para nuevos paquetes.

Para construir un paquete binario se necesita satisfacer un conjunto de dependencias de software denominadas dependencias de construcción, que se especifican en el campo `Build-Depends` del archivo `debian/control` del paquete fuente.

Si bien es posible instalar estas dependencias de software manualmente, usualmente no es deseable ya que contaminan el ambiente de construcción, por lo tanto afectando en los resultados de pruebas de instalación en sistemas nuevos.

Por ello se recomienda la utilización de software de desarrollo como `pbuilder`, que construye un ambiente nuevo e instala las dependencias de software cada vez que se requiere construir paquetes binarios. En el procedimiento anterior podrá instalar y configurar `pbuilder`.

1. Opcionalmente, descargue el paquete fuente:

⁴ Como nota especial, una vez que los procedimientos propuestos en este y otros documentos hayan sido madurados, aceptados y adoptados por la Comunidad de Desarrolladores/as de Canaima, se podrá empezar a utilizar un `base.tgz` de Canaima Estable

```
dget http://ruta.al.archivo.dsc
```

Tenga en cuenta que sólo indicar la ruta al dsc hará que dget descargue el resto de archivos del paquete fuente.

2. Extraiga el paquete fuente usando como referencia el dsc:
`dpkg-source -x paquete.dsc`
3. Realice los cambios deseados en el paquete, que se extrae en una carpeta llamada `paquete-version` en el directorio de trabajo
4. Actualice el paquete fuente para guardar sus cambios:
`dpkg-source -b paquete/`
5. Construya el/los paquete/s binario/s con sus cambios:
`pbuilder -build paquete.dsc`

Tenga en cuenta que si el software depende de la arquitectura (es decir, cualquier paquete que no indique `Architecture: all` en `debian/control`), se generarán paquetes binarios específicos para la arquitectura donde ejecute `pbuilder`⁵, usualmente `i386` o `amd64`.

Los paquetes (fuente y binario/s) resultantes de la construcción estarán en el directorio `/var/cache/pbuilder/result`. Es recomendable que mantenga limpio este directorio. Al igual que el archivo `dsc`, que referencia a todos los otros archivos del paquete fuente, el archivo `changes` que conseguirá en este directorio referencia a todos los otros archivos de los paquetes fuente y binario y por lo tanto debe preservarlo ya que este se usa para el trabajo en el repositorio.

Otros métodos para construir el/los paquete/s binario/s se consideran descontinuados para la preparación de paquetes que serán subidos al repositorio, por ejemplo: `debuild`, `debuild-pbuilder`, `dpkg-deb -b` o `dpkg -b`, `dpkg-buildpackage`, `debian/rules binary`. La razón para descontinuarlos es que todos ellos condicionan la ejecución y la construcción del software a las condiciones específicas del equipo de trabajo del desarrollador.

Inclusión y actualización de paquetes fuente y binarios a un repositorio

Se supone en este punto que dispone de un paquete fuente (`.dsc`, `.orig.tar.gz`/`.tar.gz`, `.diff.gz`) y de uno o más paquetes binarios (`.deb`) así como un archivo de cambios (`.changes`) que se generó cuando construyó los paquetes binarios (ver procedimiento anterior)

El archivo de cambios referencia a todos los paquetes y archivos relacionados, y es por ello que se utiliza para incluir, con una sola instrucción, todos los paquetes en un repositorio:

1. `reprepro include estable paquete.changes`

A futuro, otro/a desarrollador/a de Canaima GNU/Linux podría incluir soporte para otra arquitectura. Suponga que Ana utiliza `i386` e incluyó en el repositorio el paquete fuente y los paquetes

⁵ Es decir, la arquitectura de su estación de trabajo. Tenga en cuenta que esto no es lo mismo que contaminar o condicionar la construcción del paquete fuente o binario con su entorno específico ya que el paquete construido de esta forma se considera apto para enviar a granjas de construcción de cada una de las arquitecturas específicas que soporte oficial o experimentalmente Canaima (p.ej., `mipsel`, `powerpc`, `amd64`, `i386`)

binarios para i386, pero Luis, otro desarrollador, quiere preparar paquetes para amd64.

En este caso Luis descargaría el paquete fuente (usando `dget`, ver procedimiento anterior) y lo construiría con `pbuilder` en su equipo amd64. Luego sólo tendría que incluir en el repositorio los paquetes binarios (el fuente ya ha sido incluido), lo cual se describe en el *Manual de referencia para el diseño y desarrollo del sistema operativo Canaima GNU/Linux y derivados*.

Actualización de paquetes fuente y binarios originales de Debian

Los procedimientos anteriores son de amplia utilidad para todos los casos en los que el Equipo de Desarrolladores/as de Canaima desee modificar o crear paquetes de software. Por ejemplo, los paquetes *overlay* y metapaquetes *canaima-**, aplicaciones nuevas empaquetadas para Canaima o paquetes modificados para incluir/eliminar funcionalidades, todos son casos donde los procedimientos anteriores aplican y deben normalizarse.

Sin embargo, la gran mayoría de los paquetes continuarán importándose desde la distribución *stable* de Debian, por lo que es necesario normalizar un proceso para importar y actualizar los paquetes fuente y binarios que correspondan a esos paquetes que no se incluirán manualmente.

`Reprepro`, la herramienta utilizada en Canaima para mantener los repositorios, permite hacer esta actividad con facilidad utilizando el método `Update`. Con este método, se crea una lista de paquetes y sus dependencias (es decir, todos los paquetes que no se vayan a manipular en Canaima) y se le indica el repositorio fuente de descargas.

Utilizando la orden `reprepro update estable` se traerán todos los paquetes binarios y fuente, nuevos o con cambios desde los repositorios oficiales de Debian.

El/la administrador/a del repositorio es responsable por mantener al día la lista de archivos a replicar desde el repositorio oficial de Debian.

Preparación de los repositorios para una nueva versión estable de Canaima

Ante una nueva versión estable de Canaima, se debe prestar especial atención a la sanidad de los repositorios, ya que los/as usuarios/as utilizarán sus herramientas de actualización para acceder a los repositorios e instalar cualquier cambio disponible en la nueva versión estable.

1. Asegúrese de que todos los paquetes binarios modificados en Canaima tengan su paquete fuente correspondiente en el repositorio. Esta es una verificación manual -aunque automatizable- ya que se depende de que todos/as los/as desarrolladores/as que han subido sus cambios al repositorio lo hayan hecho de acuerdo a este procedimiento.
2. Asegúrese de que todos los paquetes fuente y binarios no modificados en Canaima estén actualizados con respecto a la última versión estable de Debian
`reprepro update estable`
3. Ejecute `reprepro check estable` para verificar cualquier archivo que pueda faltar.

Preparación de los medios de instalación para una nueva versión estable de Canaima

TODO (discusión en desarrolladores@c.s.g.v)

Referencias

- *Manual de referencia para el diseño y desarrollo del sistema operativo Canaima GNU/Linux y derivados*
- *Debian Policy*
- *Debian Developer's Reference*
- *Debian New Maintainer Guide*
- *Aptitude Manual*
- *Reprepro Documentation*