

Empaquetamiento de RPM

http://www.richzendy.org/docs/RPM_Taller

Que es un archivo RPM?

Un archivo rpm es formato de archivo definido para el manejador de paquetes rpm y en esencia es un archivo comprimido que contiene metadata que ofrece información como por ejemplo:

Nombre del programa.

Versión.

Datos del empaquetador.

Dependencias.

Scripts de instalación.

Ruta que deben tener los archivos que contiene en el sistema de ficheros.

Construir el Entorno

- **Hacer el entorno de construcción.**

Hacer los directorios:

- rpmdev-setuptree
- mkdir -p ~/rpmbuild/{BUILD,RPMS,SOURCES,SPECS,SRPMS}

- **Agregar la ruta a tu entorno de construcción en el archivo ~/.rpmmacros:**

- echo "%_topdir /home/`echo \$USER`/rpmbuild" >> ~/.rpmmacros

- **Instalar paquetes de construcción de rpm**

- dnf groupinstall "RPM Development Tools"
- dnf install fedora-packager
- dnf install rpmdevtools

El archivo spec

- El archivo spec define todas las reglas de compilación o construcción del paquete e incluye toda la metadata que el rpm debe contener al ser construido.
 - El comando `rpmdev-newspec` puede ser usado para crear archivos spec a partir de templates.

```
rpmdev-newspec -m -t lib -o unac.spec
```

Sección de info (spec cont.)

Name: unac

Summary: unac is a C library and command that removes accents from a string.

Version: 1.7.0

Release: 1

License: GPL

Group: Development/Libraries

URL: <http://savannah.nongnu.org/projects/unac/>

Packager: Edwind Richzendy <richzendy@fedoraproject.org>

Vendor: Richzendy Repository, <http://repo.Richzendy.org/>

Source: <http://download.savannah.gnu.org/releases/unac/unac-%{version}.tar.gz>

Patch0: unac-unactest1.diff

BuildRoot: %{_tmppath}/%{name}-buildroot

%description

unac is a C library and command that remove accents from a string.

Sección %prep (spec cont.)

Esta sección debe extraer el código fuente y aplicar cualquier parche que sea necesario para la construcción, generalmente la extracción se realiza con la macro %setup con la opción -q (quiet).

```
%prep  
%setup -q  
%patch0 -p1 -R
```

Secciones %build %install y %makeinstall (spec cont.)

Estas secciones definen todas las reglas de compilación del programa

```
%build  
%configure  
make  
%install  
%{__rm} -rf %{buildroot}  
%makeinstall
```

Secciones `%post` `%postun` (spec cont.)

La sección `%post` se usa para realizar a través de comandos o pequeños scripts configuraciones o personalizaciones que permiten una mejor adaptación del programa a la distribución.

La sección `%postun` se asegura que cualquier cambio realizado al momento de la instalación a través de las directivas en `%post` sean removidas al momento de la desinstalación y que el sistema quede completamente limpio de cualquier rastro del programa como cuando no había sido instalado.

La sección %files (spec cont.)

Lista los archivos que deben ser incluidos con el paquete, estos archivos son los que se muestran cuando el comando rpm -ql es invocado.

Esta sección hace uso extensivo de macros para referirse a los directorios y en esta sección también se puede especificar los atributos o permisos de los archivos.

Cada distribución basada en rpm usa sus propias macros.

Sección %changelog (spec cont.)

En esta sección se define el log de cambios del empaquetador, cada vez que se cree una nueva versión del paquete, esta sección debe tener una nueva entrada de log.

%changelog

*** Thu Dec 08 2009 Richzendy <richzendy@fedoraproject.org> - 1.7.0**

- Initial package.

Construcción del RPM

- Construir rpm a partir de spec
 - `rpmbuild -ba unac.spec`

rpm cumple con estándar Fedora?

Fedora provee el programa **rpmlint** para verificar si un rpm cumple con el estándar de Fedora para ser parte de la distribución.

- `rpmlint RPMS/x86_64/unac-1.7.0-1.x86_64.rpm`

Si consigue errores, una ayuda para poder corregirlos puede ser:

- https://fedoraproject.org/wiki/Common_Rpmlint_issues

Recursos

- https://docs.fedoraproject.org/en-US/Fedora_Draft_Documentation/0.1/html/RPM_Guide/index.html
- https://docs.fedoraproject.org/en-US/Fedora_Draft_Documentation/0.1/html/Packagers_Guide/index.html
- <http://richzendy.org/2009/07/15/rpm-howto-primera-parte.html>
https://fedoraproject.org/wiki/How_to_create_an_RPM_package/es
- <http://fedoraproject.org/wiki/Packaging/RPMMacros>
- <http://proyecto-fedora.org.ve>

Contacto

Edwind Richzendy Contreras Soto
richzendy@fedoraproject.org
<http://www.Richzendy.org>
irc.freenode.org @ #fedora-latam



Introducción

Este manual pretende cubrir lo básico que necesitaría para construir sus propios RPM's, sin embargo no pretende ser una documentación definitiva ni absoluta, pero si una guía que pueda encaminar a que se convierta en un empaquetador(a) para Fedora.

En esta documentación crearemos un rpm del programa **unac**, un sencillo programa hecho en C para remover acentos, si desea ver más información del programa unac, puede visitar:

<http://savannah.nongnu.org/projects/unac/>

Si desea comunicarse con el autor de esta documentación, en la última página puede conseguir los diferentes métodos de contacto.

Que es un archivo RPM?

Un archivo rpm es formato de archivo definido para el manejador de paquetes rpm y en esencia es un archivo comprimido que contiene metadata que ofrece información como por ejemplo:

- Nombre del programa.
- Versión.
- Datos del empaquetador.
- Dependencias.
- Scripts de instalación.
- Ruta que deben tener los archivos que contiene en el sistema de ficheros.

<http://www.proyecto-fedora.org>  fedora^f

RPM – Redhat Package Manager

Es una herramienta de administración de paquetes de software pensada básicamente para GNU/Linux, es capaz de instalar, actualizar, desinstalar, verificar y solicitar programas.

Los paquetes creados para funcionar con RPM (el manejador de paquetes), tienen la extensión `.rpm` y se les denomina paquetes o archivos RPM.

Un sistema GNU/Linux al estar basado en UNIX, está conformado por piezas pequeñas de software que juntas conforman un sistema complejo, dichas piezas de software están diseñadas para interactuar entre ellas y ser dependientes las unas de las otras, para ello, los programas esperan que las librerías y programas de los cuales dependen se encuentren en ubicaciones específicas dentro del sistema.

En un principio, no existían manejadores de paquetes como RPM e instalar y mantener actualizado el software era una completa pesadilla, cada quien tenía que compilar sus propios programas y a menudo habían muchos problemas de dependencias, RPM permite que alguien (un empaquetador) pueda distribuir su software en formato binario y código fuente (`.src.rpm`) y compartirlo con los demás basado en un estándar.

RPM fue originalmente desarrollado para Red Hat para Red Hat Linux en 1997 por [Erik Troan and Marc Ewing](#), sin embargo en la actualidad muchas otras distribuciones lo usan, como por ejemplo Fedora, OpenSuse, Mageia, Suse Linux, CentOS.

Construir el Entorno

- **Hacer el entorno de construcción.**
 - Hacer los directorios:
 - rpmdev-setuptree
 - mkdir -p ~/rpmbuild/{BUILD,RPMS,SOURCES,SPECS,SRPMS}
 - Agregar la ruta a tu entorno de construcción en el archivo ~/.rpmmacros:
 - echo "%_topdir /home/echo \$USER/rpmbuild" >> ~/.rpmmacros
- Instalar paquetes de construcción de rpm
 - dnf groupinstall "RPM Development Tools"
 - dnf install fedora-packager
 - dnf install rpmdevtools

http://www.proyecto-fedora.org fedora^f

El Entorno

Se crean los siguientes directorios:

BUILD BUILDROOT RPMS SOURCES SPECS SRPMS

El directorio **BUILD**, se usa para extraer y compilar el software a empaquetar y allí mismo se pueden aplicar los parches necesarios.

BUILDROOT es un directorio que en el momento de construcción del paquete RPM se enjaula para proceder a instalar el programa compilado con toda su estructura de ficheros como si estuviera en un sistema linux real.

RPMS, si no ocurre ningún error, en este directorio se colocarán los archivos rpm generados, normalmente se ordenan adentro en directorios dependiendo de la arquitectura del sistema para el que fueron creados.

SOURCES, acá va el código fuente, archivos adicionales y los parches a aplicar.

SPECS, acá va nuestro archivo .spec

SRPMS, si no ocurre ningún error, acá se colocarán los archivos RPM con las fuentes.

Los archivos **.src.rpm** son archivos que se pueden distribuir para que cualquiera pueda compilar el rpm en su propio sistema para que la compilación sea optimizada para el mismo y aplicar modificaciones de ser necesario.

El comando **dnf groupinstall "RPM Development Tools"**, instala las herramientas necesarias para construir RPM's, sin embargo dependiendo de cada paquete es posible que necesite instalar adicionalmente herramientas de compilación como por ejemplo el compilador de C, gcc, make, etc...

El archivo spec

- El archivo spec define todas las reglas de compilación o construcción del paquete e incluye toda la metadata que el rpm debe contener al ser construido.
 - El comando `rpmdev-newspec` puede ser usado para crear archivos spec a partir de templates.

```
rpmdev-newspec -m -t lib -o unac.spec
```

<http://www.proyecto-fedora.org>  fedora^f

El archivo `.spec` es un archivo de texto plano y es obra del empaquetador, podríamos decir que es un script o guión con todas las reglas necesarias para compilar y construir un archivo rpm, además de las reglas para instalar y remover la aplicación.

Luego de construido un rpm no se puede ver a partir del mismo el archivo `.spec` (excepto algunas secciones), pero todos los archivos `.src.rpm` contienen internamente un archivo `.spec` ya que se necesita para construir el rpm a partir del mismo.

Una forma de aprender a empaquetar es descargar archivos `.src.rpm` de los repositorios de Fedora que contienen este tipo de archivos, extraerlos y leer y estudiar los archivos `.spec`, si desea descargar el **.src.rpm** de algún paquete en los repositorios de Fedora, puede usar el siguiente comando:

```
dnf download --source nombre_de_paquete
```

El archivo `.spec` tiene una sintaxis específica que hay que conocer y está dividido en secciones para una mejor organización dentro del mismo.

Secciones de un archivo `.spec`

- Info
- `%prep`
- `%build`
- `%install`
- `%makeinstall`
- `%post`
- `%postun`
- `%files`
- `%changelog`

Si quiere, puede crear un archivo **.spec** vacío con el comando:

```
rpmdev-newspec -m -t lib -o unac.spec
```

Dentro del directorio **SPECS** y abrirlo con un editor de texto para revisar su contenido, el comando **rpmdev-newspec** crea una plantilla de archivo **.spec** desde el cual se puede partir y personalizar luego según las necesidades cada paquete.



Esta sección contiene toda la información del paquete, la mayoría de esta información se guarda en la base de datos RPM cuando se instala el paquete y además pasa a formar parte de la metadata que puede ser consultada de los paquetes que se encuentran en los repositorios.

Para consultar esta información de un archivo rpm sin instalar ejecute:

```
rpm -qip paquete.rpm
```

Para consultar esta información de un archivo rpm ya instalado en el sistema:

```
rpm -qi nombre-de-paquete
```

Para consultar la información de un paquete en un repositorio:

```
dnf info nombre-de-paquete
```

En la sección “Info” es importante conocer que algunas de los atributos pueden ser re-utilizados en forma de etiquetas o variables para no tener que estar cambiando por ejemplo la versión del paquete dentro del archivo con cada actualización, por ejemplo en la parte de **Source**, que por cierto se usa para declarar la ubicación desde donde se puede conseguir el código fuente, podemos observar lo siguiente:

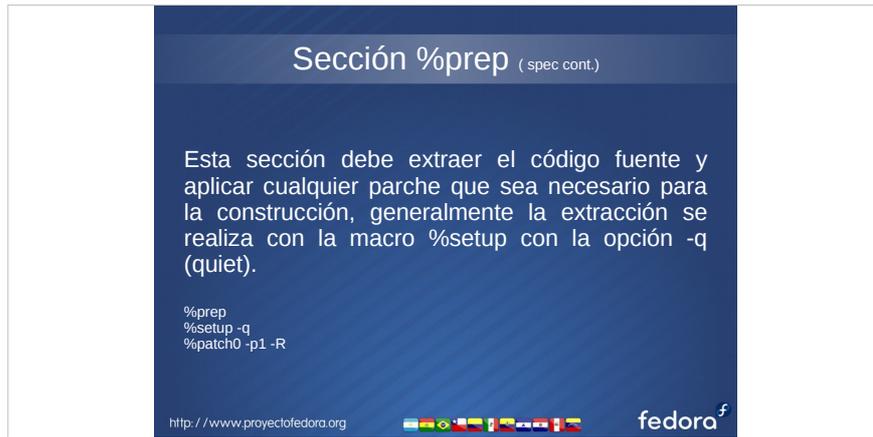
```
http://download.savannah.gnu.org/releases/unac/unac-%{version}.tar.gz
```

Bien podría ser también:

```
http://download.savannah.gnu.org/releases/unac/{name}-%{version}.tar.gz
```

Algunas cosas de esta sección pueden ser opcionales, como por ejemplo la directiva **Vendor**, o la directiva **Patch**, otras son obligatorias, por ejemplo **Version** que es donde se declara la versión del paquete que le da el creador del mismo, la directiva **Release** que viene a ser como la versión de nuestro RPM, la directiva **Name** la cual es el nombre del programa.

La directiva **Group**, indica el grupo en el que se cataloga el software y no puede ser arbitrariamente colocado, existen algunas categorías ya pre-establecidas, ver <https://fedoraproject.org/wiki/RPMGroups> o el archivo `/usr/share/doc/rpm/GROUPS`



Una de las particularidades de la construcción de un RPM, es que el código fuente que se distribuye en un archivo .src.rpm o desde el cual se parte para la construcción de los binarios debe permanecer inalterado, ya que si se necesita modificar algo del programa internamente, en el momento de la construcción del RPM dichos cambios son aplicados mediante archivos de parches.

Dicho esto, la sección %prep del archivo .spec se usa para definir las reglas de extracción del fichero con el código fuente con el comando %setup y además la aplicación de parches de ser necesario con el comando %patch, es decir, la preparación del código fuente a ser usado en la construcción del RPM.

Seguido a la instrucción %prep a menudo va el comando %setup que se considera parte de la sección %prep, con dicho comando se desempaqueta el código fuente, los modificadores más comúnmente usados de este comando son:

- **-q** : Suprime la salida innecesaria. Este es comúnmente utilizado.
- **-n nombre** : Si el archivo tarball Fuente se desempaqueta en un directorio cuyo nombre no es el nombre del RPM, este modificador puede utilizarse para especificar el nombre del directorio correcto. Por ejemplo, si el archivo tarball se desempaqueta en el directorio FOO, use «%setup -q -n FOO»
- **-c nombre** : Si el archivo tarball Fuente se desempaqueta en varios directorios en lugar de un solo directorio, este modificador puede utilizarse para crear un directorio llamado nombre y luego descomprimir en él.

Desde la versión 4.4.2 de RPM están disponibles el par de macros %{patches} y %{sources} y son útiles si usted tiene una gran lista de parches o fuentes:

```
for p in %{patches}; do
    ...
done
```

Una de las misiones de un empaquetador Fedora es distribuir con su paquete la mínima cantidad de parches de forma tal de que cuando por necesidad se tenga que usar alguno, el mismo se debe tratar de que forme parte de la siguiente versión del código fuente del programa para que no sea necesario en una próxima construcción del rpm.



Esta sección es una de las principales y es la que más puede cambiar dependiendo del tipo de software a empaquetar ya que es la sección destinada a tener las instrucciones de compilación del paquete, en la lámina se pueden ver las instrucciones básicas de compilación de un programa hecho en lenguaje C.

En el ejemplo tenemos los típicos comandos de **configure**, **make** y **make install** que son las instrucciones estándar para compilar cualquier programa en lenguaje C, la sección **build** puede ser usada para preparar el software a compilar de ser necesario y acepta instrucciones sh (comandos CLI),

En esta parte hay que prestar particular atención a la parte de **%configure**, la mayor parte de errores de construcción se dan en esta parte, lo ideal es construir el paquete con una versión del sistema operativo base para poder detectar adecuadamente las dependencias necesarias para la construcción, en Fedora podemos usar **vagrant** con una imagen base para no realizar las tareas de construcción sobre una computadora funcional y así evitar cometer algún error que destruya el sistema.

Si desea conocer información adicional sobre **vagrant**, puede revisar el siguiente enlace:

- <http://fedoramagazine.org/running-vagrant-fedora-22/>

Secciones %post %postun (spec cont.)

La sección **%post** se usa para realizar a través de comandos o pequeños scripts configuraciones o personalizaciones que permiten una mejor adaptación del programa a la distribución.

La sección **%postun** se asegura que cualquier cambio realizado al momento de la instalación a través de las directivas en %post sean removidas al momento de la desinstalación y que el sistema quede completamente limpio de cualquier rastro del programa como cuando no había sido instalado.

<http://www.proyecto-fedora.org>  fedora^f

No es obligatorio colocar algo en estas secciones, normalmente son usadas por servicios, por ejemplo, para configurar de que un servicio por defecto arranque al inicio, o que luego de que se instale arranque el servicio automáticamente.

El %postun se usa para hacer realizar alguna tarea al momento de remover el paquete, por ejemplo para remover algún archivo de configuración.

Sin embargo en Fedora no se recomienda, ni dejar servicios arrancando al inicio luego de que se instalan o remover los archivos de configuración de un servicio al desinstalar el paquete.

En nuestro caso del paquete unac, ejecutamos en el .spec ldconfig para actualizar la caché de la biblioteca de librerías compartidas del sistema al momento de instalar y remover el paquete.



En la sección **%files**, se definen las rutas que van a tener todos los ficheros contenidos en el **RPM** cuando este sea instalado en el sistema operativo.

En esta sección se hace bastante uso de las macros para definir las rutas, muchos de estos archivos provienen del código fuente, pero no necesariamente todos, algunos pueden ser añadidos por el empaquetador, por lo general el empaquetador puede añadir una página de manual (manpage), documentación adicional o algún complemento que crea conveniente para mejorar la distribución del software.

Para ver un listado de macros disponibles puede ver el siguiente enlace:

<https://fedoraproject.org/wiki/Packaging:RPMMacros>

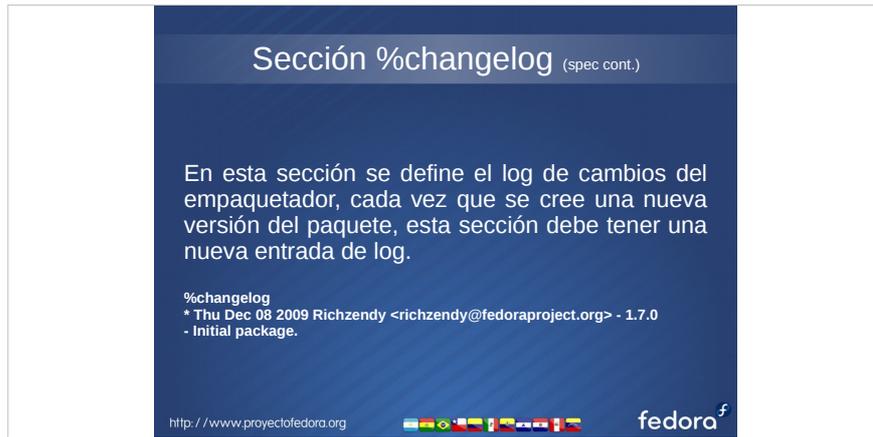
En el **.spec** del programa **unac**, tenemos lo siguiente:

```
%files
%defattr(-,root,root)
%doc ChangeLog README
%{_bindir}/*
%{_libdir}/*
%{_includedir}/*
%{_mandir}/man*/*
```

Por lo que puede ver, se pueden configurar los permisos de los archivos o ejecutables con la instrucción **%defattr**, por ejemplo usando el formato:

```
%defattr(<file mode>, <user>, <group>, <dir mode>)
```

La macro **%doc**, se refiere a la ubicación **/usr/share/doc**, **%{_mandir}** a la ubicación de páginas de manual en **/usr/share/man**, **%{_bindir}** a la ubicación **/usr/bin**



Esta sección es la única que le pertenece realmente al empaquetador y es donde el mismo agrega el histórico de cambios en su paquete, por cada versión de su paquete, se deja un par de líneas con la fecha, sus datos y un comentario de los cambios realizados.

Es de vital importancia colocar la fecha en el formato, adecuado, por ejemplo:

Mon Sep 07 2015

Ya que se hace una comprobación de la misma y en caso de estar errónea la construcción del rpm fallará, las 3 primeras letras corresponden al día en idioma inglés y la primera letra tiene que estar en mayúscula, luego las 3 siguientes letras corresponden al nombre del mes en inglés, luego el día en formato de dos dígitos y si es menor de 10, se coloca un cero adelante, luego el año en formato de 4 dígitos.



Para hacer pruebas de construcción de RPM, puede empezar por crear su entorno de construcción (lámina 3) y descargar el archivo RPM source desde la siguiente ubicación:

<http://repo.richzendy.org/unac/unac-1.7.0-1.src.rpm>

El archivo src.rpm se puede “instalar” usando el comando:

```
rpm -ivh unac-1.7.0-1.src.rpm
```

Eso coloca las fuentes en el directorio SOURCES y el spec en el directorio SPECS, luego estando dentro del directorio SPECS ya podemos intentar reconstruir el rpm con el siguiente comando:

```
rpmbuild -ba unac.spec
```

rpm cumple con estándar Fedora?

Fedora provee el programa **rpmlint** para verificar si un rpm cumple con el estándar de Fedora para ser parte de la distribución.

`rpmlint RPMS/x86_64/unac-1.7.0-1.x86_64.rpm`

Si consigue errores, una ayuda para poder corregirlos puede ser:

- https://fedoraproject.org/wiki/Common_Rpmlint_issues

<http://www.proyecto-fedora.org>  fedora^f

Fedora dispone de herramientas online para verificar y construir RPM's, sin embargo si usted desea revisar localmente si su rpm puede ser considerado válido para ser aceptado dentro de Fedora puede usar el comando **rpmlint**, el cual permite hacer una revisión de las políticas de construcción de un rpm y dar sugerencias en caso de encontrar elementos discordantes.

Algunos de estos elementos discordantes pueden ser warnings que en muchos casos se pueden obviar, pero los errores si deben ser corregidos para tener un paquete candidato.

Siempre puede apoyarse en un mentor parte de Fedora para recomendaciones y obtener ayuda para construir su paquete.

Recursos

- https://docs.fedoraproject.org/en-US/Fedora_Draft_Documentation/0.1/html/RPM_Guide/index.html
- https://docs.fedoraproject.org/en-US/Fedora_Draft_Documentation/0.1/html/Packagers_Guide/index.html
- <http://richzendy.org/2009/07/15/rpm-howto-primera-parte.html>
https://fedoraproject.org/wiki/How_to_create_an_RPM_package/es
- <http://fedoraproject.org/wiki/Packaging/RPMMacros>
- <http://proyecto-fedora.org.ve>

<http://www.proyecto-fedora.org>



fedora^f

Contacto

Edwind Richzedy Contreras Soto
richzedy@fedoraproject.org
<http://www.Richzedy.org>
irc.freenode.org @ #fedora-latam

<http://www.proyecto-fedora.org>



fedora^f