There are no translations available.
Aprende a trabajar con este gestor de paquetes desde el intérprete de comandos de la shell...

Administración de paquetes con RPM

Red Hat Package Manager es un sistema gestor de paquetes creado en origen para la distribución Red Hat aunque puede ser usada bajo cualquier otra distribución Linux y UNIX. En este artículo vamos a aprender a trabajar con este gestor de paquetes desde el intérprete de comandos de la shell.

Antes de proseguir hay que tener en cuenta el concepto de paquete. Un paquete es un archivo que contiene una aplicación o más concretamente los archivos que integrarán la aplicación (archivos de configuración, documentación y los archivos que interrelacionados componen la propia aplicación) generados al administrarse por esta herramienta.

En este documento tomaremos como ejemplo el siguiente paquete:

prog-1.0-1.i386.rpm

Nombre del paquete	prog
Versión	1.0
Lanzamiento	1
Arquitectura	i386

Gracias a esta herramienta de administración completamente gratuita se permite al usuario desempaquetar el código fuente de forma que los archivos binarios que componen el paquete pueden ser localizados e instalados, o bien reconstruir los paquetes originales con facilidad. Además, gracias a que gestiona una base de datos de todos los paquetes instalados en el sistema nos facilita el obtener información o verificar los paquetes o los archivos que derivan de estos.
Mediante esta potente herramienta el usuario podrá así pues gestionar nuestros programas de una forma óptima, rápida y sencilla: mediante el intérprete de comandos usando comandos de una sola línea.

OBJETIVOS DE RPM:

RPM presenta una serie de características:

- **Orientado a la actualización:** Nos permite actualizar programas o componentes de nuestro sistema operativo sin necesidad de reinstalar el software de nuevo.
- **Consultas poderosas:** Mediante su base de datos podremos: encontrar un paquete, encontrar un archivo, paquete de origen de un archivo específico u origen de procedencia de un paquete.
- **Verificación de Sistema:** Nos permite detectar la falta de algún archivo perteneciente a un paquete o el cambio en un archivo de configuración de un paquete.

OBTENER RPM y PAQUETES RPM:

Si no trabajamos bajo una distribución Red Hat podremos obtener la herramienta en el siguiente enlace o de los discos de instalación de nuestra distribución Red Hat pertinente.

http://www.rpm.org

Para encontrar algun paquete en específico podríamos encontrarlos en la web puesto que al ser software gratuito existen muchas fuentes en Internet para descargarlos. No obstante una buena fuente de recursos rpm serían estas:

http://www.rpmfind.com

http://www.redhat.com/download/mirror.html

USO de RPM:

Vamos a ver como gestionar paquetes con la herramienta RPM, no obstante no entraremos en detalle de cómo crear y compilar paquetes, únicamente nos centraremos en la administración de paquetes ya creados para la administración y gestión de nuestro software.

INSTALACIÓN DE RPMS

Vamos a ver como instalar un paquete. A la hora de instalar un paquete específico usaremos las opciones:

-i : Tambien notado --install, indica a RPM que instale el paquete especificado.

-U : También podría notarse --upgrade, ambas expresiones son equivalentes. Esta opción indica a RPM que queremos realizar una operación de actualización. Uno podría preguntarse porqué usar un parámetro de actualización en vez de un parámetro de instalación, como podría ser -i. La explicación es bien sencilla, al actualizar rpm nos instala el paquete deseado a la vez que nos elimina todas las posibles versiones antiguas del paquete si es que tuviesemos alguna. Es decir, con -i nos instalaría el paquete deseado pero nos podría dar problemas si tuviesemos alguna versión antigua, cosa que con -U no sucede.

-v : Indica a rpm que nos muestre por el terminal información acerca de como se va realizando el proceso de instalación.

: Indica a RPM que nos muestre por pantalla marcas "#" que nos indican el porcentaje del proceso que se ha llevado a cabo hasta el momento, para poder hacernos una idea de como se va procesando la operación.

Así pues la mejor forma de instalar un paquete evitandonos posibles situaciones de error sería:

rpm -Uvh prog-1.0-1.i386.rpm

Con esta orden se instalaría el paquete. Si todo a ido bien la salida que nos mostrará por pantalla será:

Preparing... ### [100%]

1:prog ### [100%]

De forma que primero nos muestra el nombre del paquete y las almohadillas nos irán apareciendo como medidores de progreso (gracias a las opciones añadidas **-vh**)

Si no se verificase la firma del paquete o fuese otra firma nueva (por ser por ejemplo un paquete de actualización) nos mostraría error.

La instalación de paquetes así pues es sencilla pero se nos podrían presentar una serie de **err ores:**

a) Que el paquete esté ya instalado.

Ya instalado, no que tuviesemos una version antigua.

Nos mostraría por pantalla:

Preparing... ### [100%]

Package prog-1.0-1.i386.rpm is already installed

No obstante si nos encontramos en esta situacion y queremos que nos haga la instalación ignorando el hecho de que lo tengamos ya instalado usaremos la opción **-i** para instalar en vez de actualizar a la vez que
--replacepkgs
que forzaría la instalación ignorando la existencia de una instalación.

rpm -ivh --replacepkgs prog-1.0-1.i386.rpm

Esta opción es muy util para reinstalar archivos del paquete original que se han perdido o modificado.

b) Archivos en conflicto:

Se nos puede dar la situación de que al instalar nuestro paquete en este tengamos un archivo que ya ha sido instalado por otro paquete o por una version mas antigua del mismo paquete. En este caso nos mostraría:

Preparing... ### [100%]

file /usr/bin/prog from install of prog-1.0-1 conflicts with file from the package bar-2.0.20

Para forzar a RPM a ignorar este error usaremos la opción **-i** para instalar en vez de actualizar y **--replacefiles** para forzar a que sobreescriba los archivos necesarios instalados por otro paquete.

rpm -ivh --replacefiles prog-1.0-1.i386.rpm

c) Dependencias no resueltas:

Algunos paquetes pueden depender de otros previamente instalados para funcionar correctamente. Si intentamos instalar un paquete con una dependencia no resuelta veríamos por pantalla:

Preparing... ### [100%]

error: Failed dependencies:

bar.so.2 is needed by prog-1.0-1

Suggested resolutions:

bar-2.0.20-3.i386.rpm

Para solucionar el problema deberemos obtener el paquete del que depende el que queremos instalar y una vez tengamos ambos paquetes usaremos el comando:

rpm -ivh prog-1.0-1.i386.rpm bar-2.0.20-3.i386.rpm

Y veremos por pantalla lo siguiente:

Preparing... ### [100%]

1: prog ### [50%]

2:bar ### [100%]

Si no nos sugierieran con qué paquete resolver la dependencia, mediante la opción **--redhatprovides** podríamos averiguar cual es el paquete que contiene el archivo necesario. (Es necesario tener

instalado previamente el paquete rpmdb-redhat).Para ello nos valdremos de la opción
-q
que indica a RPM que realice una consulta, opción que veremos con mas detalle mas adelante.

rpm -q --redhatprovides bar.so.2

Si el paquete que contiene el archivo se encuentra en la base de datos instalada desde rpmdb-redhat nos aparecerá el nombre del paquete:

bar-2.0.20-3.i386.rpm

Aún así, si quisiesemos tendríamos la opción de instalar el paquete sin resolver la dependencia, aunque no es aconsejable puesto que el paquete no funcionaría correctamente, con la opción **--nodeps.**

DESINSTALACIÓN DE RPMS

Se haría mediante el comando:

rpm -e prog-1.0-1

No obstante al desinstalar el paquete nos podríamos encontrar un error de dependencia ya que los archivos del paquete que estamos desinstalando dependen aparte de otro paquete. Si quisieramos desinstalarlo no obstante (no es recomendable) podríamos hacerlo añadiendo la opción --nodeps.

ACTUALIZACIÓN DE RPM

El proceso de actualización de un paquete bajo rpm tiene una mecanica bastante sencilla: básicamente lo que hace es desinstalar el paquete existente e instalando la nueva versión

posteriormente.
Para actualizar un paquete usaremos el comando:

rpm -Uvh prog-2.0-1.i386.rpm

Los posibles errores que nos podría reportar este comando serían los habituales de una desinstalación y de una instalación.

Si al actualizar un paquete nos encontramos que ya tenemos una versión mas actual del paquete a instalar y aún así queremos realizar la operación lo haríamos de la siguiente manera:

rpm -Uvh --oldpackage prog-1.0-1.i386.rpm

(suponiendo que queremos reemplazar la versión prog-2.0-1.i386.rpm)

REFRESCAMIENTO DE RPM

Si bien la opción de actualizar un paquete nos instala un paquete tengamos una versión previa o no, la opción de refrescamiento funciona de forma similiar con la excepción de que sólo instala una version nueva si ya teniamos instalada una versión anterior.

rpm -Fvh prog-2.0-1.i386.rpm

Esta opción es muy util para actualizar toda una serie de paquetes y no sabemos de cuales de ellos tenemos una versión previa instalada. Mediante el comando:

rpm -Fvh *.rpm

Actualizaríamos todos los paquetes de nuestro directorio, pero solamente se instalarían los paquetes de los que ya teniamos una versión anterior.

CONSULTAS RPM

A la hora de realizar una consulta siempre usaremos la opción -q. Así pues si usamos el comando:

rpm -q prog

Nos daría por el terminal el nombre del paquete, la versión y número del paquete instalado. Con esta opción podríamos usar una serie de Opciones de especificación de paquetes que son:

-a Nos consulta todos los paquetes instalados.
-f consultará el paquete que contenga el archivo especificado (hay que espec
-p consultara el paquete especificado.

Aparte tenemos otra opciones de especificación que nos especificarán el tipo de información a mostrar:

-i Nos mostrará información del paquete como su nombre, descripción, versión, tamaño, fecha c
-l Muestra el numero de archivos contenidos en el paquete.
-s Muestra el estado de todos los archivos del paquete.
-d Muestra una lista de archivos marcados como archivos de documentación.
-c Muestra una lista de archivos marcados como archivos de configuración.

VERIFICACIÓN DE RPM

Básicamente consiste en comparar la información de los archivos instalados con la propia información del paquete original. Especificamente nos comparará parámetros como el tamaño, los permisos, el tipo, el dueño, el grupo y la suma MD5.

Para verificar un paquete usaremos la opción:

rpm -V prog-1.0-1.i386.rpm

Para verificar un paquete que contiene un determinado archivo usaremos la opción:

rpm -Vf /home/carlos/doc1_amsn.txt

(Hay que notar la ruta completa del archivo a comprobar)

Para verificar TODOS los paquetes instalados:

rpm -Va

Para verificar un paquete instalado con un archivo de paquete RPM:

rpm -Vp prog-1.0-1.i386.rpm

Si no hubiese conflicto no habria ninguna salida tras la ejecución del comando. Si hubiese algún tipo de discrepancia la salida tendría el siguiente formato: Una cadena de ocho caracteres (una c identifica un archivo de configuración) seguido por el nombre del archivo. Cada uno de esos ocho caracteres nos inica el resultado de comparar un atributo específico de ese archivo con el valor que se le supobe a ese archivo y que está escrito en la base de datos de RPM. Los siguientes caracteres indican que pruebas no han sido pasadas:

5	MD5, suma de verificación
S	tamaño de archivo
L	enlace simbólico
T	hora de modificación del archivo

D	dispositivo
U	usuario
G	grupo
M	modo (incluye tipos de archivo y permisos)
?	archivo que no se puede leer

VERIFICAR LA FIRMA DE UN PAQUETE

Para verificar la firma de un paquete y ver si el paquete ha sido dañado o alterado usaremos el sguiente comando:

rpm -K prog-1.0-1.i386.rpm

Si nos muestra como resultado:

prog-1.0-1.i386.rpm: md5 gpg OK

Nos indicará que no ha habido problemas y el archivo no ha sido dañado en el momento de la descarga.

ENLACES DE INTERÉS:

http://www.rpm.org

http://es.tldp.org/COMO-INSFLUG/es/pdf/RPM-Como.pdf

http://www.rpm.org/max-rpm/s1-rpm-install-performing-install.html

http://www.die.net/doc/linux/man/man8/rpm.8.html

http://www.redhat.com/download/mirror.html

http://k12linux.org/netadmin/webserver.html

http://www.rpmfind.com

http://www.rpm.org/max-rpm-snapshot/