

YUM Guía General de Uso

Fundamentos

[Yum](#) :: Yellow dog Updater, Modified, software para actualizar, instalar y desinstalar paquetes en sistemas basados en RPM de forma automática.

Si bien RPM es el gestor base para los paquetes, el cálculo de las dependencias puede llevarnos a de vez cuando a jalarnos los pelos de la cabeza, de ahí la necesidad de aprender a usar YUM quién nos evitará tales malos ratos y que siempre y cuando tengamos un acceso a un repositorio YUM, podremos sonreír e instalar, desinstalar y actualizar cualquiera de los paquetes de software en nuestro Fedora o sistema basado en RPM.

La ventaja fundamental es el cálculo de las dependencias y en consecuencia su inclusión automática para poder instalar lo que queremos.

Luego tenemos a los primos gráficos GUIs gpk-application y gpk-update-viewer que son frontales de Yum para gestionar el software de nuestro sistema Fedora. gpk-application se invoca desde el menú *System > Administration > Add/Remove Software* y gpk-update-viewer desde el menú *System > Administration > Update System*. También existe otro primo GUI, yumex que es una herramienta más avanzada que ofrece más poder de control sobre yum desde el entorno X.

En cualquiera de los casos, la gestión del software es una operación sensible para el sistema y en consecuencia se requiere de privilegios suficientes para poder llevar a cabo dichas operaciones, típicamente debe ser root.

yum es una herramienta de consola, si desea concentrarse en la gestión GUI de software para empezar en Fedora, léa [PackageKit](#). En muchos casos es suficiente utilizar las herramientas GUI de alto nivel, en otros casos puede ser imposible utilizarlas, por ejemplo un servidor sin entorno GUI (X), por ello aprender a usar yum tiene doble beneficio, mayor poder y mayor alcance. Siendo justos hay que mencionar que PackageKit también incluye herramientas tipo consola en cuyo caso la discusión es si usar dichas herramientas versus la herramienta nativa.

Instalar, Actualizar, Remover

Instalar desde los repositorios configurados

```
[root@elbax ~] # yum install xfig
```

```
...
```

```
Dependencies Resolved
```

Package	Arch	Version	Repository	Size
Installing:				
xfig	i386	3.2.5-10.fc9	fedora	467 k
Installing for dependencies:				
Xaw3d	i386	1.5E-12.fc9	updates-newkey	166 k
tcsh	i386	6.15-5.fc9	updates-newkey	473 k
transfig	i386	1:3.2.5-4.fc9	updates-newkey	239 k
xfig-common	i386	3.2.5-10.fc9	fedora	4.8 M

```
Transaction Summary
```

```
Install      5 Package(s)
```

```
Update      0 Package(s)
Remove      0 Package(s)
```

```
Total download size: 6.1 M
Is this ok [y/N]:
```

En este ejemplo queremos instalar xfig pero necesitamos una serie de dependencias no instaladas previamente en nuestro sistema.

Yum se ha dado cuenta de ello y ha incluido las dependencias para que podamos tener éxito, ¿fácil no?

Error de dedo, mire la sugerencias

Es posible que al querer instalar un paquete de software no sepamos a ciencia cierta el nombre o cometamos un error de dedo en la especificación del nombre del paquete deseado, en ese caso yum incluye una búsqueda y sus resultados como sugerencias:

```
# yum install zim
...
Setting up Install Process
Parsing package install arguments
No package zim available.
* Maybe you meant: Zim
Nothing to do
```

Note la sugerencia de que tal vez lo que quiso decir era “Zim” en vez de “zim”.

Los repositorios configurados

Es importante resaltar que hablamos en esta sección de instalar paquetes a partir de los repositorios configurados, pero, ¿cuáles son?

```
$ yum repolist enabled
...
repo id                                repo name
status
fedora                                Fedora 10 - i386
enabled: 11,416
livna                                  rpm.livna.org for 10 -
i386                                  enabled:      3
rpmfusion-free                        RPM Fusion for Fedora
10 - Free                            enabled:    324
rpmfusion-free-updates                RPM Fusion for Fedora
10 - Free - Updates                  enabled:    582
updates                               Fedora 10 - i386 -
Updates                              enabled:   4,443
repolist: 16,768
```

Instalar un RPM

En el caso de que nos suministren un archivo RPM podemos utilizar el comando yum **localinstall** así:

```
# yum localinstall psad-2.1.4-1.i386.rpm
```

De esta forma si hay dependencias, yum las calculará e instalará. Una opción adicional necesaria es **--nogpgcheck** en los casos de que cree sus propios RPM y no los firme. Si el proveedor del paquete no lo firma, no es realmente recomendable que lo use a menos que tenga absoluta confianza del

origen del paquete, exija de las casas de software que firmen sus paquetes por razones de seguridad.

```
# yum --nogpgcheck localinstall psad-2.1.4-1.i386.rpm
```

Actualizar un paquete

Esta opción es más que natural, veamos el ejemplo de uso.

```
# yum update xfig
```

Actualizar con un RPM

Este es un caso similar en el que nos han suministrado un archivo RPM para actualizar.

```
# yum localupdate psad-2.1.4-1.i386.rpm
```

Actualización general

Si lo que desea es actualizar su sistema y todas sus aplicaciones instaladas use el comando siguiente:

```
# yum update
```

Remover un paquete

Para la desinstalación de software las cosas son igual de simples.

```
# yum remove xfig
```

...

Dependencies Resolved

Package	Arch	Version	Repository	Size
Removing:				
xfig	i386	3.2.5-10.fc9	installed	1.1 M
Removing for dependencies:				
xfig-common	i386	3.2.5-10.fc9	installed	11 M

Transaction Summary

```
=====
Install      0 Package(s)
Update       0 Package(s)
Remove       2 Package(s)
```

Is this ok [y/N]:

Exclusión al actualizar

De tanto en tanto, o por cualquier razón, nos interesa excluir este o cual paquete en una actualización. Para ello use la opción `--exclude`, por ejemplo así:

```
# yum update --exclude=ekiga
```

Esto puede que sea necesario porque en un momento dado los espejos que usa no se actualizan lo suficientemente rápido y no quiere esperar, o simplemente no desea usar la nueva versión.

Listar y obtener información de paquetes

Ya que en la sección anterior aprendimos a instalar, remover y actualizar software con Yum, veamos ahora un uso un poco más avanzado de Yum en la consola, veamos por ejemplo qué paquetes tenemos instalados:

```
[gomix@movix ~] $ yum list installed
```

Installed Packages

```
GConf2.i386 2.14.0-1 installed
GConf2-devel.i386 2.14.0-1 installed
GFS.i386 6.1.5-0.FC5.1 installed
GFS-kernel-smp.i686 2.6.15.1-5.FC5.33 installed
.... [muchos otros]
```

Veamos qué paquetes están disponibles para ser instalados en los repositorios configurados:

```
[gomix@movix ~] $ yum list available
```

Setting up repositories

Reading repository metadata in from local files

Available Packages

```
915resolution.i386 0.5.2-3.fc5 fedora-extras-ve
BackupPC.noarch 2.1.2-6.fc5 fedora-extras-ve
BibTool.i386 2.48-4.fc5 fedora-extras-ve
.... [muchos otros]
```

Ahora veamos qué paquetes instalados tienen actualizaciones disponibles:

```
[gomix@movix ~] $ yum list updates
```

Setting up repositories

fedora-actualizaciones-ve [1/3]

fedora-extras-ve [2/3]

fedora-core-ve [3/3]

Reading repository metadata in from local files

Updated Packages

```
apr-util.i386 1.2.8-1.fc6 fedora-actualiza
cups.i386 1:1.2.7-1.5.fc6 fedora-actualiza
cups-libs.i386 1:1.2.7-1.5.fc6 fedora-actualiza
... [otros]
```

En ciertas ocasiones nos interesa poder instalar versiones previas de cierta aplicación, no la última. En ese caso debemos primero configurar en yum.conf para que dichas versiones se muestren y poder averiguar las versiones de hecho disponibles previas ya que por omisión yum siempre lista solamente la última versión:

/etc/yum.conf

```
# agregue
showdupesfromrepos=1
```

Note como ahora se lista la versión antigua disponible en el repositorio fedora.

```
# yum list dcrw
Installed Packages
dcrw.i386                8.89-1.fc10                installed
Available Packages
dcrw.i386                8.82-1.fc9                 fedora
```

Miremos ahora un poco más profundo y consultemos los detalles de un paquete en particular con **yum info**.

```
$ yum info squid
...
Installed Packages
Name      : squid
Arch      : i386
Epoch    : 7
Version   : 3.0.STABLE13
Release   : 1.fc10
Size      : 4.3 M
Repo      : installed
Summary   : The Squid proxy caching server
URL       : http://www.squid-cache.org
License   : GPLv2+
Description: Squid is a high-performance proxy caching server for Web clients,
              : supporting FTP, gopher, and HTTP data objects. Unlike traditional
              : caching software, Squid handles all requests in a single,
              : non-blocking, I/O-driven process. Squid keeps meta data and
especiall  : hot objects cached in RAM, caches DNS lookups, supports non-
blocking   : DNS lookups, and implements negative caching of failed requests.
              :
              : Squid consists of a main server program squid, a Domain Name System
              : lookup program (dnsserver), a program for retrieving FTP data
              : (ftpget), and some management and client tools.
```

Buscar con yum

yum search nos ofrece el listado de resultados de los paquetes producto de la búsqueda por coincidencia de cadena de caracteres en la descripción, resumen y nombre del paquete rpm. Veamos su uso con un ejemplo:

```
$ yum search imap
...
===== Matched: imap
=====
dovecot.i386 : Dovecot Secure imap server
asterisk-voicemail-imap.i386 : Store voicemail on an IMAP server
balsa.i386 : Mail Client
courier-imap.i386 : Maildir-oriented IMAP4/POP3 server
cyrus-imapd.i386 : A high-performance mail server with IMAP, POP3, NNTP and
SIEVE support
cyrus-imapd-devel.i386 : Cyrus IMAP server development files
cyrus-imapd-perl.i386 : Cyrus IMAP server utility Perl modules
cyrus-imapd-utils.i386 : Cyrus IMAP server administration utilities
deletemail.i386 : A non-interactive tool for deleting mails
fetchmail.i386 : A remote mail retrieval and forwarding utility
getmail.noarch : POP3, IMAP4 and SDPS mail retriever with Maildir delivery
horde-enhanced.noarch : Adds extra functionality to the horde framework
imapsync.noarch : Tool to migrate email between IMAP servers
```

imp.noarch : The Internet Messaging Program: webmail access to IMAP/POP3 accounts
 ingo.noarch : The Horde web-based Email Filter Rules Manager
 isync.i386 : Tool to synchronize IMAP4 and Maildir mailboxes
 libc-client.i386 : UW C-client mail library
 libsieve.i386 : A library for parsing, sorting and filtering your mail
 libsieve-devel.i386 : A library for parsing, sorting and filtering your mail
 mailx.i386 : Enhanced implementation of the mailx command
 nagios-plugins-tcp.i386 : Nagios Plugin - check_tcp
 nginx.i386 : Robust, small and high performance http and reverse proxy server
 offlineimap.noarch : Powerful IMAP/Maildir synchronization and reader support
 perl-Mail-IMAPClient.noarch : An IMAP Client API
 perl-Net-IMAP-Simple.noarch : Perl extension for simple IMAP account handling.
 php-imap.i386 : A module for PHP applications that use IMAP
 php-pear-Net-Sieve.noarch : Communication with timsieved
 phpwapmail.noarch : WAP-based e-mail client
 pop-before-smtp.noarch : Watch log for POP/IMAP auth, notify MTA to allow relay
 python-twisted-mail.i386 : A Twisted Mail library, server and client
 roundcubemail.noarch : Round Cube Webmail is a browser-based multilingual IMAP client
 squirrelmail.noarch : SquirrelMail webmail client
 stunnel.i386 : An SSL-encrypting socket wrapper
 up-imapproxy.i386 : University of Pittsburgh IMAP Proxy
 uw-imap.i386 : UW Server daemons for IMAP and POP network mail protocols
 uw-imap-devel.i386 : Development tools for programs which will use the UW IMAP library
 uw-imap-static.i386 : UW IMAP static library
 uw-imap-utils.i386 : UW IMAP Utilities to make managing your email simpler
 wammu.noarch : Mobile Phone Manager - Gammu GUI
 xfce4-mailwatch-plugin.i386 : Mail Watcher plugin for the Xfce panel

yum provides o **yum whatprovides** es otra forma de buscar en la base de datos de paquetes. En este caso se busca qué paquete provee cierta funcionalidad o archivo. Use un nombre específico o sintáxis tipo file-glob-syntax wildcards para listar los paquetes disponibles o instalados que proveen la funcionalidad o archivo buscado.

```

$ yum provides /etc/httpd/conf/httpd.conf
...
httpd-2.2.10-2.i386 : Apache HTTP Server
Repo                : fedora
Matched from:
Filename            : /etc/httpd/conf/httpd.conf

httpd-2.2.10-2.i386 : Apache HTTP Server
Repo                : installed
Matched from:
Other               : Provides-match: /etc/httpd/conf/httpd.conf
  
```

Grupos

Ya que hemos revisado varias de las opciones del programa yum, ahora revisaremos una que no es muy popular y que ciertamente es muy útil, y es la capacidad de Yum de manejar grupos de paquetes de software para instalar, actualizar, remover o listar.

Listar grupos disponibles

```

$ su
<contraseña>
  
```

```
# yum grouplist
...
Setting up Group Process
...
Installed Groups:
  Administration Tools
  Authoring and Publishing
  Base (RPM Fusion Free)
  Base (RPM Fusion Nonfree)
  Clustering
  DNS Name Server
  Development Libraries
  Development Tools
  Editors
  Educational Software
  Engineering and Scientific
  FTP Server
  Fonts
  GNOME Desktop Environment
  GNOME Desktop Environment (RPM Fusion Free)
  Games and Entertainment
  Graphical Internet
  Graphics
  Hardware Support
  KDE (K Desktop Environment)
  Legacy Fonts
  Legacy Network Server
  Legacy Software Development
  Mail Server
  Miscellaneous command line tools (RPM Fusion Free)
  MySQL Database
  Network Servers
  Office/Productivity
  Printing Support
  Server Configuration Tools
  Sound and Video
  Sound and Video (RPM Fusion Free)
  System Tools
  Text-based Internet
  Web Server
  Window Managers
  Windows File Server
  X Window System
Available Groups:
  Fedora Eclipse
  Fedora Packager
  GNOME Software Development
  Games and Entertainment (RPM Fusion Free)
  Games and Entertainment (RPM Fusion Nonfree)
  Hardware Support (RPM Fusion Free)
  Hardware Support (RPM Fusion Nonfree)
  Internet (RPM Fusion Free)
  Java Development
  KDE (K Desktop Environment) (RPM Fusion Free)
  KDE Software Development
  News Server
  PostgreSQL Database
  SUGAR Desktop Environment
  System Tools (RPM Fusion Free)
  Virtualization
  Web Development
  X Software Development
```

XFCE
XFCE Software Development
Done

Información de Paquetes de Grupo

Para obtener información de los grupos y de sus paquetes constitutivos use `groupinfo`.

```
# yum groupinfo "Virtualization"
...
Setting up Group Process
...
Group: Virtualization
Description: These packages provide a virtualization environment.
Mandatory Packages:
    python-virtinst
Default Packages:
    kvm
    qemu
    virt-manager
    virt-viewer
Optional Packages:
    gnome-applet-vm
    virt-top
    xen
```

Instalación, Actualización y Remoción de Grupo

Digamos que deseamos comenzar a desarrollar en KDE.

```
# yum groupinstall "KDE Software Development"
.... otros tantos mas ....
kdelibs i386 6:3.5.5-0.2.fc6 fedora-ve 18 M
kdenetwork i386 7:3.5.5-0.1.fc6 fedora-ve 12 M
kdepim i386 6:3.5.5-0.2.fc6 fedora-ve 19 M
kdeutils i386 6:3.5.5-0.1.fc6 fedora-ve 3.8 M
kdnssd-avahi i386 0.1.3-0.1.fc6 fedora-ve 43 k
kdnssd-avahi-devel i386 0.1.3-0.1.fc6 fedora-ve 11 k
pcre-devel i386 6.6-1.1 fedora-ve 175 k
sip-devel i386 4.4.5-3 fedora-ve 15 k
```

Transaction Summary

=====

Install 24 Package(s)

Update 0 Package(s)

Remove 0 Package(s)

Total download size: 121 M

Is this ok [y/N] :

Obviamente averiguar todos y cada uno de los nombres de paquetes e instalarlos uno a uno, aún con sus dependencias calculadas con yum hubiera sido un dolor de cabeza.

Igual puede actualizar un grupo dado previamente instalado con por ejemplo:


```
# yum groupupdate "MySQL Database"
```

Para la remoción, tenemos la opción de usar **groupremove**, por ejemplo, puede desinstalar KDE completo con una sola instrucción:

```
# yum groupremove "KDE (K Desktop Environment)"
```

Configuración de YUM

La configuración general de YUM se encuentra en `/etc/yum.conf` y los repositorios YUM configurados en Fedora se encuentran declarados en `/etc/yum.repos.d/`. Para comprender las configuraciones partiremos de analizar estos archivos una vez que nuestro Fedora esté instalado. Para usos más avanzados agregaremos nuevas secciones y en caso de no encontrar lo que busca consulte `man yum.conf`.

`/etc/yum.conf`

```
[main]
cachedir=/var/cache/yum
keepcache=0
debuglevel=2
logfile=/var/log/yum.log
exactarch=1
obsoletes=1
gpgcheck=1
plugins=1
metadata_expire=1800
installonly_limit=2
tsflags=republish
```

```
# PUT YOUR REPOS HERE OR IN separate files named file.repo
# in /etc/yum.repos.d
```

La primera cosa que podemos notar de los archivos de configuración es su estructura de bloques de secciones con `[sección]`. Veremos que esta estructura se repite al momento de configurar los repositorios. En este caso `yum.conf` en la sección `[main]` define el comportamiento general de yum y establece las directivas base, por ejemplo, de dónde se encuentran el cache. Comentamos a abajo un poco cada una de las directivas.

cachedir

Directorio donde yum debe almacenar su cache y archivos db. El valor por omisión es `'/var/cache/yum'`.

keepcache

Ya sea `'1'` o `'0'`. Determina si yum debe o no mantener el cache de los encabezados y paquetes después de una instalación exitosa. Valor por omisión es `'1'` (mantener los archivos)

debuglevel

Nivel de mensajes de depuración en la salida. El rango práctico es de 0-10. El valor por omisión es `'2'`.

logfile

Ruta completa al nombre de archivo donde yum debe escribir sus actividades.

exactarch

Ya sea `'1'` o `'0'`. Puesto a `'1'` provoca que yum update solo actualice los paquete de arquitectura que haya instalado, es decir, con esta opción habilitada yum no instalará un paquete i686 para actualizar un paquete i386. Por omisión el valor es `'1'`.

obsoletes

Esta opción sólo tiene efecto durante una actualización. Habilita la lógica de procesamiento de obsolescencias yum. Es útil cuando se realiza actualizaciones de nivel de distribución (upgrade). Vea también la documentación del comando yum upgrade para más detalles.

Opción en la línea de comando: `--obsoletes`

gpgcheck

Ya sea '1' o '0'. Esto le indica a yum si debe o no realizar una verificación de firma GPG en los paquetes. Cuando está habilitado en la sección [main], establece el comportamiento por omisión para todos los repositorios. Esta opción también determina si una instalación de un archivo local RPM será verificado por su firma GPG o no. El valor por omisión es '0'.

plugins

Ya sea '0' o '1'. Conmutador global para habilitar o deshabilitar las extensiones (plugins) de yum. Por omisión su valor es '0' (extensiones deshabilitadas). Vea la sección PLUGINS del manual yum(8) para más información acerca de la instalación de extensiones yum.

metadata_expire

Duración en segundos después del cual los metadatos caducarán. Así entonces si los metadatos actuales descargados son menos segundos de antiguo, entonces yum no actualizará dichos metadatos a partir del repositorio. Si encuentra que yum no está descargando la información para actualizaciones tan frecuente como usted quisiera, disminuya el valor de este parámetro. También puede cambiar del formato por omisión en segundos a días, horas o minutos agregando d, h o m respectivamente al valor especificado. El valor por omisión es 1.5 hours, para complementar yum-updatesd ejecutándose una vez por hora. También es posible usar la palabra "never" implicando que los metadatos nunca expirarán.

installonly_limit

Cantidad de paquetes listados en installonlypkgs que se deben mantener instalados al mismo tiempo. Poner este valor a 0 deshabilita la funcionalidad. Por omisión el valor es '3'.

tsflags

Lista separado por comas o espacios de banderas de transacción que serán pasadas a rpm. Estas incluyen 'noscripts', 'notriggers', 'nodocs', 'test', y 'repackage'. Puede poner todas o cualquiera de ellas. Sin embargo, si no sabe qué hacen éstas banderas en el contexto de una transacción rpm, mejor no toque. Por omisión la lista está vacía.

Estas son sólo algunas directivas, para más, man yum.conf. Dentro de yum.conf puede incluir la configuración de los repositorios pero esa forma de configuración está en desuso y se prefiere utilizar archivos individuales en /etc/yum.repos.d/ que comentaremos más adelante.

/etc/yum.repos.d/*repo

/etc/yum.repos.d/fedora.repo

```
[fedora]
name=Fedora $releasever - $basearch
failovermethod=priority
baseurl=http://download.fedoraproject.org/pub/fedora/linux/releases/
$releasever/Everything/$basearch/os/
#mirrorlist=http://mirrors.fedoraproject.org/mirrorlist?repo=fedora-
$releasever&arch=$basearch
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora-$basearch

[fedora-debuginfo]
name=Fedora $releasever - $basearch - Debug
failovermethod=priority
#baseurl=http://download.fedoraproject.org/pub/fedora/linux/releases/
```

```

$releasever/Everything/$basearch/debug/
mirrorlist=http://mirrors.fedoraproject.org/mirrorlist?repo=fedora-debug-
$releasever&arch=$basearch
enabled=0
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora-$basearch

[fedora-source]
name=Fedora $releasever - Source
failovermethod=priority
#baseurl=http://download.fedoraproject.org/pub/fedora/linux/releases/
$releasever/Everything/source/SRPMS/
mirrorlist=http://mirrors.fedoraproject.org/mirrorlist?repo=fedora-source-
$releasever&arch=$basearch
enabled=0
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora-$basearch

```

Cada sección [nombre] define un repositorio yum, luego dentro de cada definición de repositorio puede sobrescribirse el comportamiento general de yum con directivas de alcance global para el manejo en particular de este repositorio o agregarse nuevas directivas específicas de alcance repositorio.

```

[fedora] << identificador de repositorio
name=Fedora $releasever - $basearch
failovermethod=priority
baseurl=http://download.fedoraproject.org/pub/fedora/linux/releases/
$releasever/Everything/$basearch/os/
#mirrorlist=http://mirrors.fedoraproject.org/mirrorlist?repo=fedora-
$releasever&arch=$basearch
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora-$basearch

```

El identificador de repositorio debe ser único.

name

Una cadena (nombre) de caracteres que describen el repositorio para el humano.

baseurl

Debe ser un URL que apunta al directorio donde residen el directorio yum ‘repodata’. Puede ser un URL tipo http://, ftp:// o file://. Usted puede especificar múltiples URLs en una declaración, la mejor forma de hacerlo es así:

```

[repositoryid]
name=Algún nombre para este repositorio
baseurl=url://server1/ruta/al/repositorio/
        url://server2/ruta/al/repositorio/
        url://server3/ruta/al/repositorio/

```

Puede utilizar autenticación básica HTTP prefijando “usuario:contraseña@” al nombre del servidor en la línea baseurl, por ejemplo “baseurl=http://usuario@passwd@example.com/”.

mirrorlist

Especifica un URL que apunta a un archivo que contiene un lista de baseurls. Puede utilizar esta declaración en vez de baseurl.

enabled

Puede ser ‘1’ o ‘0’. Esto le indica a yum de si usar o no este repositorio.

gpgcheck

Puede ser ‘1’ o ‘0’. Esto le indica a yum si debe o no realizar una verificación de firma GPG

en los paquetes obtenidos desde este repositorio.

gpgkey

Un URL que apunta a un archivo que contiene la llave ASCII de GPG para el repositorio. Esta opción se utiliza si yum necesita una llave pública para verificar los paquetes y la llave requerida no ha sido importada en la base de datos RPM. Si se coloca esta opción yum automáticamente importará la llave especificada por el URL. yum le preguntará antes por su confirmación para instalar la llave a menos que haya especificado `assumeyes`. Igualmente se pueden listar múltiples URLs en la misma forma que se hizo para `baseurl`.

failovermethod

Puede ser `'roundrobin'` o `'priority'`. `'roundrobin'` selecciona aleatoriamente un URL de la lista de URLs para comenzar y proceder luego con otro seleccionado de igual forma si falla el contacto con el servidor seleccionado. `'priority'` empieza en orden por el primer `baseurl` listado y los utiliza secuencialmente en caso de falla al intentar contactar al servidor. Si no se especifica, el valor por omisión es `'priority'`.

Configuración con Prioridad

Siguiendo con el uso avanzado de Yum, ahora veremos como podemos configurar nuestro Fedora para que use nuestros repositorios locales en cierto orden con prioridad sin descartar la lista de espejos externos. Sin más vea la configuración necesaria.

```
[fedora-core-ve]
name=Fedora Core $releasever - $basearch - Base
baseurl=http://mirepo.org/$releasever/$basearch/os/

http://otrorepo.org/$releasever/$basearch/os/

mirrorlist=http://mirrors.fedoraproject.org/mirrorlist?repo=core-
$releasever&arch=$basearch
failovermethod=priority
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora
```

Primero note el uso de múltiples URL en `baseurl`. Debe hacerse como lo he puesto aquí, no utilice múltiples declaraciones `baseurl`, no funcionará adecuadamente.

Luego note el uso de la declaración `failovermethod`, este parámetro puede tener los valores `'roundrobin'` o `'priority'`. Por omisión el valor es `'roundrobin'` que aleatoriamente selecciona un URL de la lista de URLs. El valor puesto a `'priority'` hace que se comience por el primer URL de la lista de URLs. Aquí es importante notar que la lista de URLs se construye con los `baseurl` y luego de la lista obtenida de `mirrorlist`. Así entonces logramos lo que queríamos, buscamos primeros en nuestros repos, luego si fallan, en ese orden, se busca en los URL de la lista de espejos.

Para Atrás (rollback)

NOTA: En Fedora 10 se introdujo RPM 4.6.0, que remueve la capacidad de hacer el `RollBack`.

Ya no es tan común, sobre todo si ya es un administrador experimentado, pero puede pasar que deseemos echar atrás lo que hicimos, esto se denomina en inglés `rollback`. En ciertos entornos esto es mandatorio. Después de largo tiempo usando sistemas RPM finalmente y desde hace no mucho está disponible las facilidades necesarias en rpm y en yum para poder echar atrás de forma automática y simple. Esto aún no permite tener varias versiones del mismo software, que yo sepa,

en la base de datos rpm, es uno de los pocos requerimientos que no se maneja aún en rpm hasta donde sé.

Veamos entonces qué debemos hacer para lograr el poder echar para atrás con yum/rpm.

1. Para configurar yum para que salve la información de rolback agregue la siguiente línea a su `/etc/yum.conf`.

```
tsflags=reppackage
```

1. Para configurar la utilidad de línea de comando rpm para que haga lo mismo, agregue la siguiente línea en `/etc/rpm/macros`.

```
%_reppackage_all_erasures 1
```

1. Instale, elimine y actualice a su voluntad, ya sea usando PackageKit, yumex, yum, rpm, o el servicio automático de actualizaciones yum.
2. Si y cuando desee echar atrás a un estado previo, realice una actualización rpm con la opción `--rollback` seguida de un especificador de fecha/hora. Algunos ejemplos son:

```
# rpm -Uhv --rollback '9:00 am'
# rpm -Uhv --rollback '4 hours ago'
# rpm -Uhv --rollback 'december 25'
```

La capacidad de echar para atrás es de gran valor para empresas que operan bajo condiciones estrictas como las de telecomunicaciones, en donde se debe trabajar en condiciones de “ventanas de tiempo de mantenimiento”.

También ocurre que a veces hacemos actualizaciones sin muchas evaluaciones previas y llegamos a un punto donde la mejor solución simplemente es volver al estado anterior a la actualización.

No olvide que una actualización puede haber provocado decenas de actualizaciones de dependencias. El hacer el procedimiento manualmente puede tomarle simplemente mucho tiempo.

Note que no hay un comando yum para hacer el rollback, debe hacerse con rpm. Tampoco sé que se pueda hacerlo con algún gui de yum como pirut, pup, yumex o gnome-yum.

installonly_limit

Debido a la naturaleza de cierto software, en particular los kernel, se requiere que se puedan tener instalados mas de una versión a la vez por diferentes razones, los siguientes parámetros son de relevancia en esos casos. En el pasado esta funcionalidad se manejaba a través de una extensión **installonly**, ahora en `/etc/yum.conf`:

```
installonlypkgs
```

Lista de paquetes que deben ser instalados siempre, nunca actualizados. Los kernel en particular caen en esta categoría. Por omisión el valor es `kernel, kernel-smp, kernel-bigmem, kernel-enterprise, kernel-debug, kernel-unsupported`.

```
installonly_limit
```

Cantidad de los paquetes listados en `installonlypkgs` que se deben mantener instalados al mismo tiempo. Poner a 0 deshabilita esta facilidad. Por omisión es '2'.

Extensiones Yum

El sitio para la centralización de configuración de las extensiones en Fedora es

/etc/yum/pluginconf.d. En esta guía comentamos algunas extensiones, para mucha mas información acerca de otras extensiones vea [YUM Plugins](#) o <http://wiki.linux.duke.edu/YumPlugins>

downloadonly

Instalación, obvio que con yum.

```
# yum install yum-downloadonly
...
Dependencies Resolved
=====
Package Arch Version Repository Size
=====
Installing:

yum-downloadonly noarch 1.0.1-1.fc6 fedora-extras 5.9 k

Transaction Summary
=====
Install 1 Package(s)
Update 0 Package(s)
Remove 0 Package(s)
Total download size: 5.9 k
Is this ok [y/N] : y
```

Veamos cómo podemos usarlo y recordemos que podemos usar Yum para obtener información general de los paquetes con yum info, aún si no están instalados:

```
# yum info yum-downloadonly
...
Name : yum-downloadonly
Arch : noarch
Version: 1.0.1
Release: 1.fc6
Size : 3.1 k
Repo : installed
Summary: Yum plugin to add downloadonly command option
Description:
This plugin adds a --downloadonly flag to yum so that yum will only download the
packages and not install/update them.
```

Para los que no le pegan al inglés, una de las razones de esta publicación, la descripción lo dice todo, se agrega la opción `--downloadonly` a Yum. Intentemos usarla con un paquete no instalado (note la línea de carga del módulo, ‘Loading “downloadonly” plugin’).

```
# yum --downloadonly yum-kernel-module
```

Esto nos dará un error porque yum espera un comando después de la opción. Así que ahora lo intentaremos correctamente con:

```
# yum install --downloadonly yum-kernel-module
...

Dependencies Resolved
=====
Package Arch Version Repository Size
=====
Installing:

yum-kernel-module noarch 1.0.1-1.fc6 fedora-extras-ve 7.0 k
```

Transaction Summary

=====

Install 1 Package(s)
Update 0 Package(s)
Remove 0 Package(s)
Total download size: 7.0 k
Is this ok [y/N] : y

Downloading Packages:

(1/1): yum-kernel-module- 100% |=====| 7.0 kB 00:00

exiting because --downloadonly specified

Note ahora que yum termina. Ahora la pregunta es, ¿dónde está nuestro archivo rpm? La respuesta es que depende de la configuración de yum, pero la configuración de caja almacena todo en la carpeta /var/cache/yum/<repositorio>. Como en nuestro caso el archivo rpm descargado provenía de Fedora Extras, el nombre del directorio donde se almacenó es /var/cache/yum/extras.

```
# cd /var/cache/extras
# ls
cachecookie headers packages primary.xml.gz
primary.xml.gz.sqlite repomd.xml
yum-kernel-module-1.0.1-1.fc6.noarch.rpm
```

Ahí está nuestro rpm descargado y podemos confirmar que no se ha instalado con :

```
# rpm -q yum-kernel-module
package yum-kernel-module is not installed
```

¿Qué pasa si el paquete tiene dependencias?

```
# yum --downloadonly install zaptel-utils
```

```
...
```

Dependencies Resolved

=====

Package	Arch	Version	Repository	Size
---------	------	---------	------------	------

=====

Installing:

zaptel-utils	i386	1.4.0-9.fc6	fedora-extras	32 k
--------------	------	-------------	---------------	------

Installing for dependencies:

zaptel-lib	i386	1.4.0-9.fc6	fedora-extras	13 k
------------	------	-------------	---------------	------

Transaction Summary

=====

Install 2 Package(s)
Update 0 Package(s)
Remove 0 Package(s)
Total download size: 45 k
Is this ok [y/N] : y

Downloading Packages:

(1/2): zaptel-lib-1.4.0-9 100% |=====| 13 kB 00:00

(2/2): zaptel-utils-1.4.0 100% |=====| 32 kB 00:01

exiting because --downloadonly specified

Nuevamente yum hace lo correcto, descarga todo, pero no los instala.

```
# ls
yum-kernel-module-1.0.1-1.fc6.noarch.rpm
zaptel-lib-1.4.0-9.fc6.i386.rpm
zaptel-utils-1.4.0-9.fc6.i386.rpm
```

Así que con esta extensión podemos entonces descargar los paquetes y sus dependencias para por ejemplo un computador que no tiene conexión a internet.

¿Qué pasa si el paquete ya está instalado?

```
# rpm -q httpd
httpd-2.2.3-5

# yum --downloadonly install httpd
...
Nothing to do
```

Y tiene sentido, ya está instalado. ¿Hay manera de forzar? No. Hablaremos de otro programa para resolver esta situación más adelante.

fastestmirror

Esta extensión está diseñada para ordenar la lista de espejos por velocidad de respuestas de cada uno antes de comenzar una descarga.

```
# yum install yum-fastestmirror
...
Dependencies Resolved
=====
Package Arch Version Repository Size
=====

Installing:

yum-fastestmirror noarch 1.0.2-2.fc6 fedora-extras-ve 9.9 k

Transaction Summary
=====
Install 1 Package(s)
Update 0 Package(s)
Remove 0 Package(s)
Total download size: 9.9 k
Is this ok [y/N] : y

Downloading Packages:

(1/1): yum-fastestmirror- 100% |=====| 9.9 kB 00:00

Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction

Installing: yum-fastestmirror ##### [1/1]
Installed: yum-fastestmirror.noarch 0:1.0.2-2.fc6
Complete!
```

Su configuración está en /etc/yum/pluginconf.d/fastestmirror.conf.


```
[main]
enabled=1
verbose=0
socket_timeout=3
hostfilepath=/var/cache/yum/timedhosts.txt
maxhostfileage=10
```

Al intentar una operación yum, siempre ocurre este ordenamiento. Podemos medir su efectividad desactivando y activando la extensión y realizando la misma descarga y procurando que no se use el cache.

```
## Limpieza del cache
# yum clean packages
Loading "fastestmirror" plugin
Loading "downloadonly" plugin

5 packages removed

## Deshabilito fastestmirror en su archivo de config primero, descargo de
fedora-ve.org
## Escogemos algun rpm a descargar con unos +/-10Megas para que se note la diff
## También agregamos -y para evitar el retardo de contestar de forma interactiva
# time yum -y --downloadonly install abiword
...
Totalc download size: 9.1 M
...
real 5m6.770s
user 0m11.469s
sys 0m1.660s

## Limpiamos el cache otra vez
# yum clean packages
Loading "fastestmirror" plugin
Loading "downloadonly" plugin

5 packages removed

## Vuelvo habilitar fastestmirror en su archivo de config
# yum -y --downloadonly install yumex
...
real 5m23.901s
user 0m11.926s
sys 0m1.845s
```

Curiosamente, de hecho no va más rápido. No conozco los detalles de cómo mide esta extensión la velocidad de los repositorios, pero lo cierto es que no parece muy efectivo. Aún así y de acuerdo a los experimentos que realicé, si funciona de acuerdo a su propia lógica, es decir, sea como sea que mide los servidores, usa el mejor de acuerdo a esa medida, de ahí que si hay posibilidades de que nos vaya mejor. Una de las pruebas que hice fue verificar que esta extensión actúa después de armar la lista de servidores sin importar el método de failovermethod seleccionado en la configuración de los repositorios, es decir que funciona sin importar el orden de dicha lista, se vuelve a ordenar de acuerdo a la medida de velocidad y se escoge el “más rápido”.

Utilidades yum-utils

Existen una serie de utilidades que se encuentran disponibles en Fedora en el paquete denominado yum-utils. Esta sección está dedicada a dicho paquetes y los programas que allí se incluyen.

yumdownloader

Este utilitario complementa al plugin downloadonly ya que dicho plugin sólo funciona si el software deseado no está instalado. Veamos como falla y cómo podemos resolver con yumdownloader.

```
$ yum --downloadonly install openssh-server
Loading "downloadonly" plugin
Setting up Install Process
Setting up repositories
Reading repository metadata in from local files
Parsing package install arguments
Nothing to do
```

Pues claro, no hay nada que instalar, ya está instalado. Intentemos ahora con yumdownloader.

```
$ yumdownloader openssh-server
openssh-server-4.3p2-19.f 100% |=====| 252 kB    00:01
```

Listo, ya tenemos el rpm que queríamos.

package-cleanup

Este utilitario ayuda a detectar problemas en la base de datos rpm y a resolverlos.

Usos comunes:

```
# Listar paquetes huérfanos, aquellos paquetes que no son provistos por ninguno
de los repositorios configurados,
#"son huérfanos de repositorio".
$ package-cleanup --orphans
```

```
# Listar los paquetes con duplicados en la base de datos RPM.
$ package-cleanup --dupes
```

```
...
samba-3.2.8-0.27.fc10.i386
samba-3.2.8-0.26.fc10.i386
libtalloc-1.2.0-26.fc10.i386
libtalloc-1.2.0-27.fc10.i386
wireshark-gnome-1.0.6-1.fc10.i386
wireshark-gnome-1.0.5-1.fc10.i386
```

```
# Listar los problemas de dependencias en la base de datos RPM.
$ package-cleanup --problems
```

```
...
Reading local RPM database
Processing all local requires
Missing dependencies:
Package samba requires samba-common = 0:3.2.8-0.26.fc10
Package wireshark-gnome requires wireshark = 1.0.5-1.fc10
```

Veamos algunos ejemplos.

```
# package-cleanup --orphans
Setting up yum
libsndfile-1.0.14-1.fc5.i386
grisbi-0.5.8-2.fc5.i386
gedit-2.14.1-1.i386
...
```

NOTAS DE USO: debido a la forma en como se maneja las actualizaciones, package-cleanup sólo

considera la última versión del paquete, los paquetes no actualizados pero provistos por repositorios configurados aparecerán como huérfanos, es decir, el repositorio Updates, por ejemplo, no continua suministrando la versión anterior para efectos de package-cleanup. Es decir, antes de hacer un package-cleanup --orphans, se recomienda hacer una actualización completa si es posible, tenga cuidado de no remover estos paquetes a la ligera, ya que también pueden ser paquetes instalados manualmente y que funcionan perfectamente, o simplemente se deshabilito el repositorio desde el cual se hizo la instalación.

Con `package-cleanup` y el uso de la opción `--duplicates`, podemos detectar los paquetes duplicados en la base de datos RPM, en este ejemplo `gedit-2.14.1-1.i386`.

```
# package-cleanup --duplicates
...
gedit
...
# rpm -qa | grep gedit
gedit-2.14.1-1
gedit-2.14.3-1
```

Tenemos dos versiones instaladas y la versión antigua permaneció, probablemente por algun mal manejo. De ahí que podemos sin problemas comenzar una tarea de limpieza, y es que de eso se trata `package-cleanup`.

```
# yum remove gedit-2.14.1-1
...
Setting up Remove Process
Resolving Dependencies
--> Populating transaction set with selected packages. Please wait.
---> Package gedit.i386 1:2.14.1-1 set to be erased
--> Running transaction check
```

Dependencies Resolved

Package	Arch	Version	Repository	Size
Removing:				
gedit	i386	1:2.14.1-1	installed	13 M

Transaction Summary

```
=====
Install      0 Package(s)
Update       0 Package(s)
Remove       1 Package(s)
Is this ok [y/N] : y
Downloading Packages:
Running Transaction Test
```

Después de realizar esta operación hemos recuperado cierto espacio en disco efectivamente hemos limpiado, al menos en parte, nuestra base de datos rpm que representa al software instalado en nuestro sistema Fedora. Para más detalles de uso revise el manual, **man package-cleanup**.

yum-complete-transaction

Si hay transacción yum incompletas `yum-complete-transaction` le ofrece la oportunidad de terminarlas en orden. Típicamente cuando esta situación es necesaria el comando yum de turno le dará una salida parecida a:

```
...
```

There are unfinished transactions remaining. You might consider running yum-complete-transaction first to finish them.

...

yum-complete-transaction es un program que busca transacciones yum incompletas o abortadas en el sistema e intenta completarlas. Busca en los archivos transaction-all* y transaction-done* que normalmente se encuentran en /var/lib/yum si una transacción ha sido abortada en el medio de su ejecución.

Si se encuentra más de una transacción no terminada se intentará completar la más reciente primero. Puede ejecutar este programa más de una vez para limpiar todas las transacciones sin terminar.

```
$ sudo yum-complete-transaction
```

...

```
There are 1 outstanding transactions to complete. Finishing the most recent one
The remaining transaction had 3 elements left to run
Package recode-3.6-26.fc9.i386 already installed and latest version
Package fortune-mod-1.99.1-11.fc9.i386 already installed and latest version
Package gtypist-2.7-6.fc9.i386 already installed and latest version
Nothing in the unfinished transaction to cleanup.
Cleaning up completed transaction file
```

repo-graph

Uso:

```
repo-graph [options]
```

Opciones:

-h, --help	show this help message and exit
--repoid=REPOID	specify repositories to use
-c CONFFILE	config file location

repo-rss

repo-rss es un programa para generar flujos RSS para uno o más repositorios Yum. Vea los detalles de su uso en **man repo-rss**.

repoclosure

Lee los metadatos de una serie de repositorios y verifica todas las dependencias en todos los paquetes y su resolución. Imprime una lista de paquetes con dependencias no resueltas.

Uso:

```
repoclosure [-c <config file>] [-a <arch>] [-r <repoid>] [-r <repoid2>]
```

repomanage

Gestiona un directorio de paquetes rpms. Devuelve listas de paquetes recientes o antiguos en un directorios para poder pasar fácilmente via pipes a xargs o programas similares.

Uso:

```
repomanage [--old] [--new] path.
```

repoquery

repoquery es un programa para consultar información en los repositorios YUM, éstas son similares a las consultas rpm.

Un ejemplo simple, listar el contenido de un paquete no instalado.

```
$ repoquery -l yum-tsflags
/etc/yum/pluginconf.d/tsflags.conf
/usr/lib/yum-plugins/tsflags.py
/usr/lib/yum-plugins/tsflags.pyc
/usr/lib/yum-plugins/tsflags.pyo
## Note que no está instalado
$ rpm -q yum-tsflags
package yum-tsflags is not installed
```

Para más detalles vea la página del manual al respecto, **man repoquery**.

reposync

reposync se utiliza para sincronizar un repositorio remoto yum con un directorio local utilizando yum para recuperar los paquetes.

Uso:

```
reposync [options]
```

Opciones:

Options:

```
-h, --help                show this help message and exit
-c CONFIG, --config=CONFIG config file to use (defaults to /etc/yum.conf)
-a ARCH, --arch=ARCH      act as if running the specified arch (default: current
                           arch, note: does not override $releasever)
-r REPOID, --repoid=REPOID specify repo ids to query, can be specified multiple
                           times (default is all enabled)
-e CACHEDIR, --cachedir=CACHEDIR
                           directory in which to store metadata
-t, --tempcache            Use a temp dir for storing/accessing yum-cache
-d, --delete              delete local packages no longer present in repository
-p DESTDIR, --download_path=DESTDIR
                           Path to download packages to: defaults to current dir
-g, --gpgcheck             Remove packages that fail GPG signature checking after
                           downloading
-u, --urls                 Just list urls of what would be downloaded, don't
                           download
-n, --newest-only          Download only newest packages per-repo
-q, --quiet                Output as little as possible
-l, --plugins              enable yum plugin support
```

repotrack

repotrack se mantiene al tanto de cualquier paquete y sus dependencias. Descargará el o los paquetes que desea monitorear y todas sus dependencias.

Uso:

```
repotrack [options] package1 [package2] [package..]
```

Opciones:

```
-h, --help          show this help message and exit
-c CONFIG, --config=CONFIG
                    config file to use (defaults to /etc/yum.conf)
-a ARCH, --arch=ARCH check as if running the specified arch (default:
                    current arch)
-r REPOID, --repoid=REPOID
                    specify repo ids to query, can be specified multiple
                    times (default is all enabled)
-t, --tempcache     Use a temp dir for storing/accessing yum-cache
-p DESTDIR, --download_path=DESTDIR
                    Path to download packages to
-u, --urls          Just list urls of what would be downloaded, don't
                    download
-n, --newest        Toggle downloading only the newest packages(defaults
                    to newest-only)
-q, --quiet         Output as little as possible
```

yum-builddep

yum-builddep es un programa que instala los RPMs necesarios para construir el paquete especificado. El SRPM del paquete especificado debe estar disponible en el repositorio Yum o puede ser un archivo SRPM local.

```
# yum-builddep sox
```

```
...
```

Package	Arch	Version	Repository	Size
Installing:				
alsa-lib-devel	i386	1.0.18-6.rc3.fc10	fedora	996 k
gsm-devel	i386	1.0.12-6.fc9	fedora	12 k
libvorbis-devel	i386	1:1.2.0-5.fc10	fedora	356 k

Transaction Summary

Install	2 Package(s)
Update	0 Package(s)
Remove	0 Package(s)

Total download size: 368 k

Is this ok [y/N]:

yum-groups-manager

yum-groups-manager se usa para crear o editar el archivo de metadata de grupos para un repositorio yum. Esto es usualmente mucho más fácil que editar el XML a mano. yum-groups-manager puede cargar un archivo entero de metadatos de grupo o crear uno nuevo o editar un grupo existente y luego escribir los metadatos como salida.