

Fedora 14

Software Management Guide

The definitive guide for software management



Guillermo Gómez Savino

Fedora 14 Software Management Guide

The definitive guide for software management

Edition 1.0

Author

Guillermo Gómez Savino

guillermo.gomez@gmail.com

Copyright © 2010 Red Hat, Inc. and others.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at <http://creativecommons.org/licenses/by-sa/3.0/>. The original authors of this document, and Red Hat, designate the Fedora Project as the "Attribution Party" for purposes of CC-BY-SA. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, MetaMatrix, Fedora, the Infinity Logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

For guidelines on the permitted uses of the Fedora trademarks, refer to https://fedoraproject.org/wiki/Legal:Trademark_guidelines.

Linux® is the registered trademark of Linus Torvalds in the United States and other countries.

Java® is a registered trademark of Oracle and/or its affiliates.

XFS® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

All other trademarks are the property of their respective owners.

This manual is dedicated to software management in Fedora. The reader will learn the skills necessary to be able to install, remove, and update software in his or her system. This manual is designed for all technical levels: from users of Fedora desktop, to system administrators and Fedora developers.

Preface	ix
1. Document Conventions	ix
1.1. Typographic Conventions	ix
1.2. Pull-quote Conventions	x
1.3. Notes and Warnings	xi
2. We Need Feedback!	xi
1. Introduction	1
1.1. Purpose	1
1.2. Audience	1
1.3. Manual Organization	1
2. Introduction to software management	3
2.1. RPM	4
2.2. Software Repositories	4
2.3. YUM	5
2.4. PackageKit	5
3. The gnome-packagekit GUI	7
3.1. gpk-application	7
3.1.1. Application State	7
3.1.2. Listings	8
3.1.3. Package Installation	10
3.1.4. Package Removal	13
3.1.5. Listing Filters	14
3.2. gpk-log	17
3.3. gpk-repo	17
3.4. gpk-update-viewer	18
3.5. gpk-update-icon	19
3.6. gpk-prefs	19
4. YUM Configuration	21
4.1. General Configuration	21
4.2. YUM Variables	22
4.3. Cache Management	23
4.4. Priority Configuration	24
4.5. Bandwidth Control	25
4.6. Proxy Configuration	25
4.7. Multiarchitecture Policy	25
4.8. Management of YUM Repositories	26
4.8.1. Add New Repositories	26
4.8.2. Manual Configuration of a Repository	26
4.8.3. Removing Repositories	28
4.8.4. The Fedora Repositories	29
4.8.5. Third Party and Personal Repositories	30
4.8.6. List of Mirrors and Metalink	30
4.8.7. Configuration for Package Protection	32
5. Yum Command Usage	33
5.1. Yum Commands and Syntax	33
5.2. Packages names	35
5.3. The Help Command	36
5.4. Repository Listing	36
5.5. Installing New Software	37
5.5.1. Install from the enabled repositories	37
5.5.2. Finger error, look at the suggestions	39

5.5.3. Installing an RPM	39
5.5.4. Reinstalling Software	40
5.6. Updating Software	40
5.6.1. Updating a Package	41
5.6.2. Updating with an RPM	41
5.6.3. Downgrade to a Previous Version	41
5.6.4. General Update	42
5.6.5. Checking and Update Listing	42
5.6.6. Exclude when Updating	42
5.6.7. Avoiding packages with problems when updating	43
5.7. Removing Software	43
5.8. Querying Package Information	44
5.9. Queries and Package Listings	45
5.9.1. Search by Name, Description and Summary	45
5.9.2. Search by file or functionality	45
5.9.3. Listing installed packages	46
5.9.4. Listing of available packages	46
5.9.5. Listing available updates	46
5.9.6. Listing recent packages	46
5.9.7. Listing foreign packages	47
5.10. Enable a repository on the go	47
5.11. Dependencies Listing	48
5.12. Query the cache	48
5.13. Show duplicates	49
5.14. Existing lock	49
5.15. YUM Software Groups	50
5.15.1. Listing available groups	50
5.15.2. Packages info from a group	51
5.15.3. Installing and Removing Groups	52
5.16. YUM history and logs	53
5.16.1. History	54
5.16.2. Undo? Redo!	54
5.16.3. Corrupt History	55
5.16.4. Responsible Identification with sudo	56
5.16.5. Checking the rpm database	57
6. Yum extensions	59
6.1. Extension installation and general configuration	59
6.2. yum-plugin-aliases	60
6.2.1. Configuration	60
6.2.2. Usage examples	61
6.3. yum-plugin-auto-update-debug-info	63
6.3.1. Configuration	63
6.3.2. Usage examples	63
6.4. yum-plugin-changelog	64
6.4.1. Configuration	64
6.4.2. Usage examples	64
6.5. yum-plugin-downloadonly	65
6.5.1. Configuration	65
6.5.2. Usage examples	65
6.5.3. What happens with the package dependencies?	67
6.5.4. What happens with packages already downloaded?	67
6.6. yum-plugin-fastestmirror	67
6.6.1. Configuration	68

6.6.2. Usage examples	68
6.7. yum-plugin-filter-data	69
6.7.1. Configuration	69
6.7.2. Documentation	69
6.7.3. Usage examples	69
6.8. yum-plugin-fs-snapshot	71
6.8.1. Configuration	71
6.8.2. Documentation	72
6.9. yum-plugin-keys	72
6.9.1. Configuration	72
6.9.2. Usage examples	72
6.10. yum-plugin-list-data	75
6.10.1. Documentation	76
6.10.2. Configuration	76
6.10.3. Usage examples	76
6.11. yum-plugin-local	77
6.11.1. Configuration	77
6.11.2. Usage examples	78
6.12. yum-plugin-merge-conf	80
6.12.1. Configuration	81
6.12.2. Usage examples	81
6.13. yum-plugin-post-transaction-actions	82
6.13.1. Configuration	82
6.13.2. Usage examples	82
6.14. yum-plugin-priorities	83
6.14.1. Configuration	83
6.14.2. Usage examples	84
6.15. yum-plugin-protectbase	84
6.15.1. Configuration	84
6.15.2. Usage examples	84
6.16. yum-plugin-refresh-updatesd	85
6.16.1. Configuration	85
6.17. yum-plugin-remove-with-leaves	85
6.17.1. Configuration	86
6.17.2. Usage examples	86
6.18. yum-plugin-warm-cache	87
6.18.1. Configuration	87
6.19. yum-plugin-security	87
6.19.1. Configuration	87
6.19.2. Usage examples	88
6.20. yum-plugin-show-leaves	90
6.20.1. Configuration	90
6.20.2. Usage examples	90
6.21. yum-plugin-tmprepo	91
6.21.1. Configuration	91
6.21.2. Usage example	91
6.22. yum-plugin-tsflags	92
6.22.1. Configuration	92
6.22.2. Usage examples	93
6.23. yum-plugin-upgrade-helper	93
6.23.1. Configuration	93
6.24. yum-plugin-verify	94
6.24.1. Configuration	94
6.24.2. Documentation	95

6.24.3. Usage examples	95
6.25. yum-plugin-versionlock	96
6.25.1. Configuration	96
6.25.2. Usage examples	96
6.26. yum-presto	97
6.26.1. Configuration	97
6.26.2. Usage examples	98
6.27. PackageKit-yum-plugin	100
6.27.1. Configuration	100
6.28. yum-langpacks	100
6.28.1. Configuration	100
6.29. Yum command usage and its extensions	100
6.29.1. Disable extensions	100
6.29.2. Usage examples	101
7. Yum Utils	103
7.1. yum-updateonboot	103
7.1.1. Configuration	103
7.2. yumdownloader	103
7.3. package-cleanup	104
7.3.1. Usage Examples	105
7.4. yum-complete-transaction	106
7.5. yum-builddep	107
7.6. yum-NetworkManager-dispatcher	108
7.7. yum-cron	109
7.7.1. Start and Stop	109
7.7.2. Configuration	109
8. Advanced YUM repository management	111
8.1. Repository creation with createrepo	111
8.2. Query repositories with repoquery	113
8.3. Groups management with yum-groups-manager	115
8.4. Creating YUM mirrors with reposync	117
8.5. Web browsable repositories with repoview	119
8.6. Unresolved dependencies with repoclosure	120
8.7. Package tracking with repotrack	121
9. PackageKit CLI	123
9.1. pkcon	123
9.1.1. Documentation	123
9.1.2. Searches	124
9.1.3. Install, update and remove packages	124
9.1.4. Packages Info	126
9.1.5. General Update	127
9.1.6. Available Updates	128
9.1.7. Groups	128
9.1.8. Repositories	129
9.1.9. Downloading Packages	130
9.1.10. Use of Filters	130
9.1.11. Transactions	130
9.2. gpk-install-local-file	131
9.3. gpk-install-catalog	132
9.4. pkmon	133
10. Rpm management elements	135
10.1. Why avoid RPM?	135

10.2. Documentation in the package	135
10.3. Configuration files in the package	136
10.4. File ownership	136
10.5. Database backup and rebuild	136
10.6. Importing keys	136
10.7. List imported keys	136
11. Techniques for system administrators	139
11.1. Remote configuration	139
11.2. syslog support	139
A. Revision History	141
Index	143

Preface

1. Document Conventions

This manual uses several conventions to highlight certain words and phrases and draw attention to specific pieces of information.

In PDF and paper editions, this manual uses typefaces drawn from the [Liberation Fonts](https://fedorahosted.org/liberation-fonts/)¹ set. The Liberation Fonts set is also used in HTML editions if the set is installed on your system. If not, alternative but equivalent typefaces are displayed. Note: Red Hat Enterprise Linux 5 and later includes the Liberation Fonts set by default.

1.1. Typographic Conventions

Four typographic conventions are used to call attention to specific words and phrases. These conventions, and the circumstances they apply to, are as follows.

Mono-spaced Bold

Used to highlight system input, including shell commands, file names and paths. Also used to highlight keycaps and key combinations. For example:

To see the contents of the file **my_next_bestselling_novel** in your current working directory, enter the **cat my_next_bestselling_novel** command at the shell prompt and press **Enter** to execute the command.

The above includes a file name, a shell command and a keycap, all presented in mono-spaced bold and all distinguishable thanks to context.

Key combinations can be distinguished from keycaps by the hyphen connecting each part of a key combination. For example:

Press **Enter** to execute the command.

Press **Ctrl+Alt+F2** to switch to the first virtual terminal. Press **Ctrl+Alt+F1** to return to your X-Windows session.

The first paragraph highlights the particular keycap to press. The second highlights two key combinations (each a set of three keycaps with each set pressed simultaneously).

If source code is discussed, class names, methods, functions, variable names and returned values mentioned within a paragraph will be presented as above, in **mono-spaced bold**. For example:

File-related classes include **filesystem** for file systems, **file** for files, and **dir** for directories. Each class has its own associated set of permissions.

Proportional Bold

This denotes words or phrases encountered on a system, including application names; dialog box text; labeled buttons; check-box and radio button labels; menu titles and sub-menu titles. For example:

Choose **System** → **Preferences** → **Mouse** from the main menu bar to launch **Mouse Preferences**. In the **Buttons** tab, click the **Left-handed mouse** check box and click

¹ <https://fedorahosted.org/liberation-fonts/>

Close to switch the primary mouse button from the left to the right (making the mouse suitable for use in the left hand).

To insert a special character into a **gedit** file, choose **Applications** → **Accessories** → **Character Map** from the main menu bar. Next, choose **Search** → **Find...** from the **Character Map** menu bar, type the name of the character in the **Search** field and click **Next**. The character you sought will be highlighted in the **Character Table**. Double-click this highlighted character to place it in the **Text to copy** field and then click the **Copy** button. Now switch back to your document and choose **Edit** → **Paste** from the **gedit** menu bar.

The above text includes application names; system-wide menu names and items; application-specific menu names; and buttons and text found within a GUI interface, all presented in proportional bold and all distinguishable by context.

Mono-spaced Bold Italic or ***Proportional Bold Italic***

Whether mono-spaced bold or proportional bold, the addition of italics indicates replaceable or variable text. Italics denotes text you do not input literally or displayed text that changes depending on circumstance. For example:

To connect to a remote machine using ssh, type **ssh *username@domain.name*** at a shell prompt. If the remote machine is **example.com** and your username on that machine is john, type **ssh *john@example.com***.

The **mount -o remount *file-system*** command remounts the named file system. For example, to remount the **/home** file system, the command is **mount -o remount */home***.

To see the version of a currently installed package, use the **rpm -q *package*** command. It will return a result as follows: ***package-version-release***.

Note the words in bold italics above — *username*, *domain.name*, *file-system*, *package*, *version* and *release*. Each word is a placeholder, either for text you enter when issuing a command or for text displayed by the system.

Aside from standard usage for presenting the title of a work, italics denotes the first use of a new and important term. For example:

Publican is a *DocBook* publishing system.

1.2. Pull-quote Conventions

Terminal output and source code listings are set off visually from the surrounding text.

Output sent to a terminal is set in **mono-spaced roman** and presented thus:

```
books      Desktop  documentation  drafts  mss    photos  stuff  svn
books_tests Desktop1  downloads      images  notes  scripts svgs
```

Source-code listings are also set in **mono-spaced roman** but add syntax highlighting as follows:

```
package org.jboss.book.jca.ex1;

import javax.naming.InitialContext;
```

```

public class ExClient
{
    public static void main(String args[])
        throws Exception
    {
        InitialContext iniCtx = new InitialContext();
        Object          ref    = iniCtx.lookup("EchoBean");
        EchoHome        home   = (EchoHome) ref;
        Echo            echo    = home.create();

        System.out.println("Created Echo");

        System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));
    }
}

```

1.3. Notes and Warnings

Finally, we use three visual styles to draw attention to information that might otherwise be overlooked.

Note

Notes are tips, shortcuts or alternative approaches to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.

Important

Important boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring a box labeled 'Important' will not cause data loss but may cause irritation and frustration.

Warning

Warnings should not be ignored. Ignoring warnings will most likely cause data loss.

2. We Need Feedback!

If you find a typographical error in this manual, or if you have thought of a way to make this manual better, we would love to hear from you! Please submit a report in Bugzilla: <http://bugzilla.redhat.com/bugzilla/> against the product **Fedora Documentation**.

When submitting a bug report, be sure to mention the manual's identifier: *software-management-guide*

If you have a suggestion for improving the documentation, try to be as specific as possible when describing it. If you have found an error, please include the section number and some of the surrounding text so we can find it easily.

Introduction

This manual is the product of the experience and evolution of the software management tools in Fedora. The contents expressed here are specific to Fedora, but many of them are easily applicable to other GNU/Linux RPM-based systems. A few chapters are independent from the distribution, such as the contents dedicated to PackageKit and its tools.

1.1. Purpose

This manual attempts to consolidate all of the relevant information regarding software management in Fedora, for all levels, from novice to expert. This manual should serve more knowledgeable people as a general reference on the subject, while for the novices it should serve as a learning guide for which they can use the first chapters to resolve their most basic needs.

1.2. Audience

- Desktop User
- Advanced User
- GNU/Linux Fedora System Administrator
- Fedora Developers and Packagers

1.3. Manual Organization

1. General introduction to the concepts of general software management.
2. General software management with graphical desktop tools (GUI).
3. The YUM system and its configuration.
4. Gaining power in the command line with YUM (CLI).

This manual is organized in a format that first presents the fundamental concepts, next the desktop tools, and finally the console tools and the more advanced techniques on the software management subject.

This manual was designed that way so that newcomers can read straight through, or can limit themselves to the first sections of the manual to understand enough necessary for a home desktop user. If you need more than the graphical desktop tools, you can continue reading through and deepen the understanding of the configuration and use of the YUM system until you become an expert in software management in Fedora.

As the reader advances through the sections, the level of complexity and difficulty also advances, scaling from the management of software of a home system, to a network of managed workstations and Fedora servers.

Introduction to software management

Every informatic system needs a way to install and remove the software it will need to do his work. But what is software installation? Installing software is just to putting the binary executable files and documentation for this software in the right places at the target system.

The right places to install software in Fedora are ruled by the Linux Filesystem Hierarchy Standard (FHS).

Once you have decided where the software must be installed, then we can choose the method to do the installation:

- By hand, copying the files with tools like `cp`, `mv`, `scp`.
- Using a tool likely to build software, for example, `make`.
 - Typically used on installation from sources.
 - `./configure ; make ; make install`
- Using some generic tools for software distribution.
 - `gem install rubygem`
 - `cpan> install DateTime`
- Using a Version Control System like `svn`, `cvs`, `git`, etc.
 - `git clone ...`
 - `svn co ...`
 - `cvs co ...`
- Using tools suited for the distribution.

This guide is about the last option and in particular Fedora, although many of the concepts, tools and techniques in this manual can be applied to other systems. The following highlights some problems when using other methods.

1. Several tools for the same task: `gem` or `cpan`?
2. It's hard to know exactly which software is installed.
3. It's tricky to do updates to installed software.
4. It doesn't scale: it becomes progressively harder at the time you install more and more software.
5. Hard or practically impossible to know who and when installed a specific software.
6. It becomes messy and tends to make systems dirty.
7. Altered paths and no certainty than you are using the software you really need.

There are little benefits of using manual methods or platform independent software management tools like:

1. Support for multiple versions.
2. You can get always the last releases from upstream.

3. Choice to proprietary software with no free and open source replacement.

Generally it is not recommended to use non-native Fedora tools for managing software as the software you need is available in RPM format, even if it is proprietary software. Also these alleged benefits are very questionable and in fact we insist that you should not use these methods unless an very special circumstance and you know in advance what you are doing.

In Fedora and like other Linux distributions today, the software management occurs at three levels which we introduce below:

Table 2.1. Three layers in software management.

Layer	Component	Comment
1	RPM	Low level, basic handling of dependencies.
2	YUM	Automatic dependencies and repository resolution.
3	PackageKit	User Graphical Interface and available in several distros.

2.1. RPM

The RPM format is the way to encapsulate all the software of a given application, or general data, for the RPM software management system existing in Fedora and other Linux distributions and Unix systems. The RPM system is both the file format and the software management system based on low-level records in a database.

So then, the chosen application can be provided by developers in Fedora RPM format compatible. It may also be provided by the Fedora Project itself if the software has been previously packaged properly and legally validated as part of the distribution. A third option is available in foreign repositories that offer the application in a RPM format package.

With RPM you can install, uninstall and update software packages with simple commands. RPM maintains a database of installed packages and there are commands for querying and system checking.

2.2. Software Repositories

The repository definition is critical, they can defined as a collection of ordered, sorted and available software to be used by compatible tools to download it and manipulate it. These repositories have data and metadata to be able to process e.g. remote searches without the need to trace full paths.

The repository concept is crucial in the history of GNU/Linux distributions, from which Fedora is one of them. What is know as "the distribution" or simply Fedora, is the DVD or LiveCD, and this perception is wrong. Fedora is a repository based distribution and it is not limited to what is chosen to build a selection and its distribution in a phisical media or particular access method. Fedora are all the software packages available in official repositories. The DVD or LiveCD for the current version of Fedora is just a particular selection and format ready to use.

Software repositories are then the base for the distribution as such but it is also a tool for system administrators to deploy their systems and networks in a more orderly, controlled and safe way. Each system administrator can build software repositories to deploy the required software to a network with workstations or servers.

You can also use it as a cache or mirror of the official repositories and avoid the excessive use of Internet for software installing and updating on Fedora workstations and servers network. Later we will stop in the creation of software repositories.

It is important to note that the repositories used in Fedora are YUM repositories that can be available via `http://` , `ftp://` and `file://` access methods.

2.3. YUM

YUM¹: Yellow dog Updater, Modified, command line interface (CLI) software to update, install and remove RPM-based packages, in an automatic and easy way.

While `rpm` is the low level manager and base for package management, the calculation of dependencies is very basic and can easily lead to a situation known as "the nightmare of the dependencies". This is the fundamental task solved by YUM system, hence to learn to use `yum` avoids many bad times and as long as you have access to a YUM repository, you can smile and install, uninstall and update any software packages in our RPM based system without complications due to dependencies.

The key advantage is the calculation of the dependencies and thus its automatic inclusion in order to be able to install the desired software package, however, it is not limited to that and from its initial design to the present days, it has gained a lot of extensions to allow the replacement of the `rpm` command line tool.

There are also GUI programs such as `gpk-application` and `gpk-update-viewer` as a YUM front-end for software management in our RPM based system, like Fedora, Red Hat Enterprise Linux, or CentOS. These applications are part of the suite called PackageKit.

`gpk-application` in Fedora is invoked from **System > Administration > Add / Remove Software** . There is another more advance GUI tool called Yumex which offers more advanced control over YUM in an X environment.

In either case, the software management is a sensitive operation for the system and therefore it requires enough privileges to perform such transactions, typically you must be root.

`yum` is the main command line tool of the YUM system. In many cases it is enough to use the high-level GUI tools, in some others it may be insufficient or even impossible to use them, such as a server without a GUI environment (X), so to learn to use the `yum` tool has a double benefit, more power and broader scope. To be fair we have to mention that PackageKit also includes command line tools, and knowing that the discussion is whether to use these tools or the `yum` native tool. The documentation for that command line client is really small and it is still not recommended to use them and prefer the use of `yum` native tool.

2.4. PackageKit

PackageKit² is a system designed to install and update software on your computer easily. It aims to unify all the graphical tools (GUIs) used in the various GNU/Linux distributions.

PackageKit is a system for software management for multiple platforms using native tools in each one of them and it is not intended to replace the native tools but to create a new layer of abstraction in the task of software management between different types of systems and to facilitate the task for users and administrators.

¹ <http://linux.duke.edu/projects/yum/>

² <http://www.packagekit.org/>

Chapter 2. Introduction to software management

According to its description PackageKit is a D-Bus abstraction layer that allows the user to manage packages securely using a multi-distro API.

This manual will describe PackageKit from the perspective of Fedora being a system which manages software with yum. With PackageKit and its graphics applications you can meet the basic and essential tasks of software management in your Fedora system.

PackageKit is the standard GUI system in Fedora for the software management, so the next installation note is in the exceptional case that you do not have PackageKit installed.

```
$ su -c "yum install PackageKit gnome-packagekit"
```

The gnome-packagekit GUI

Normally gnome-packagekit graphic users interface (GUI) are installed by default in a Fedora desktop, but if it is not, install it with root privileges doing the following:

```
# su -c "yum install gnome-packagekit"
```

3.1. gpk-application

gpk-application is the GNOME GUI standard software manager in Fedora since version 9 and looks like the screenshot shown below. gpk-application is used to install, remove and search for packages in Fedora systems.

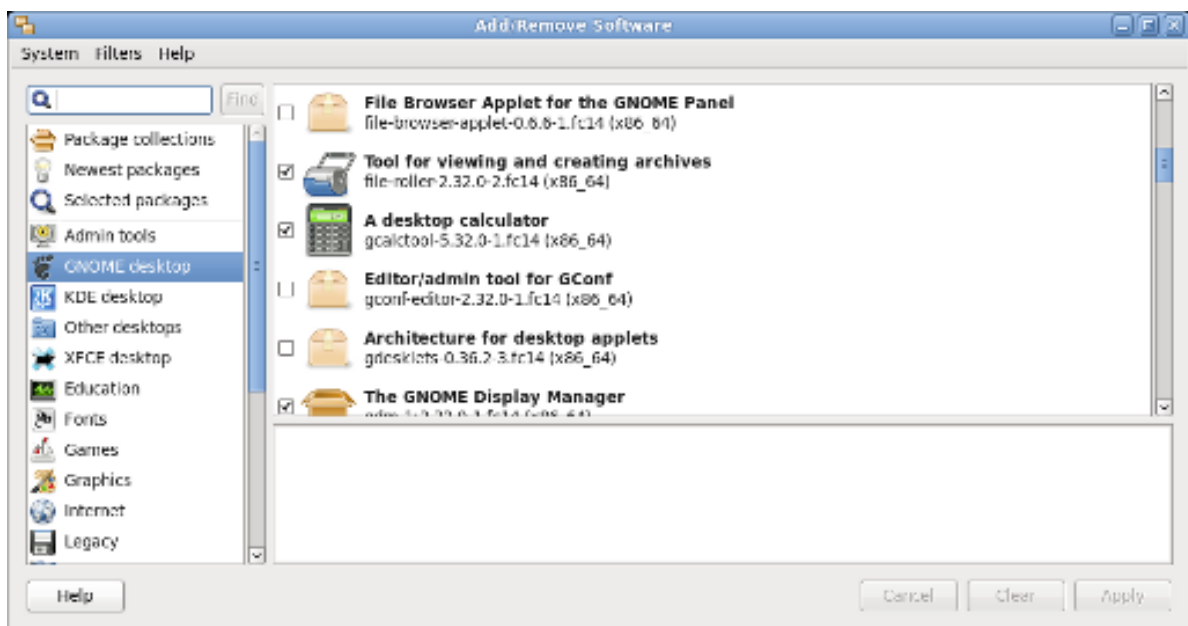


Figure 3.1. gpk-application

If it is invoked from **System** → **Administration** → **Add/Remove Software**.

3.1.1. Application State

Before detailing the operations that gpk-application can perform, it is important to note the state of the system in terms of software management, that is, the system must query to external repositories to get an updated listing of the available applications, and then, we have to wait to that operation completion, whether it succeeds or fails, and do not despair doing click everywhere because it will only create new transactions that will be queued. Below there is a screenshot highlighting in red rectangle the application status.

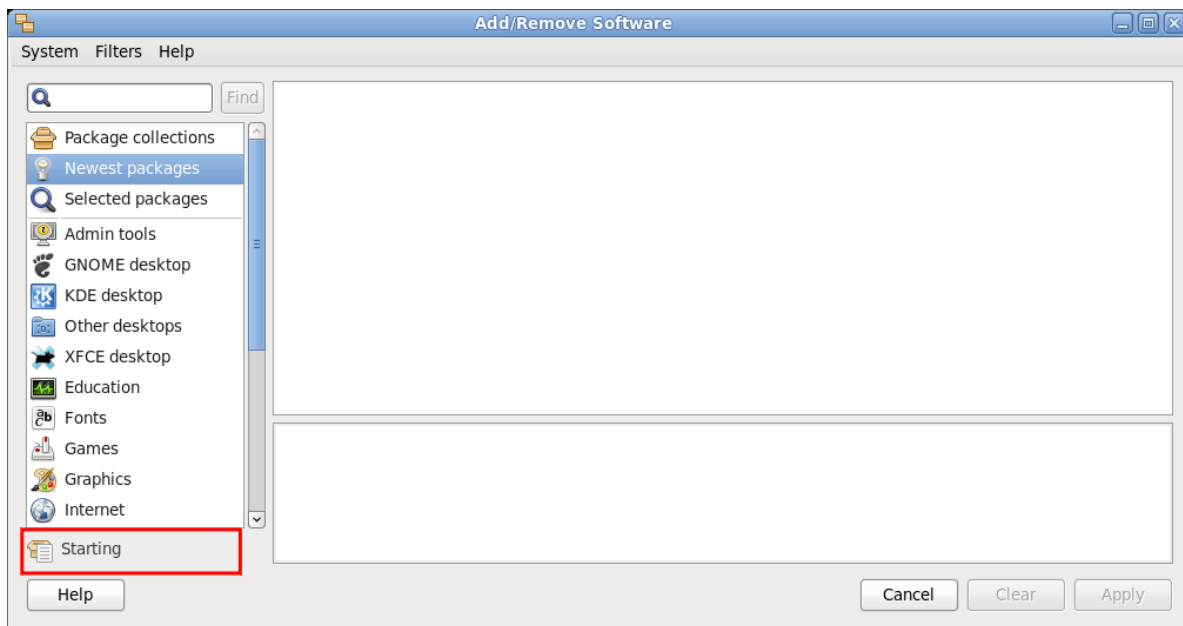


Figure 3.2. gpk-application status

3.1.2. Listings

In the application main window you can see three panels, the first one with a red frame is to select the desired package listing, the second one, with a blue frame, is to show the package listing including the package state, if it is checked then it is installed, the third panel with green frame is used to show details from the selected package in the second panel.

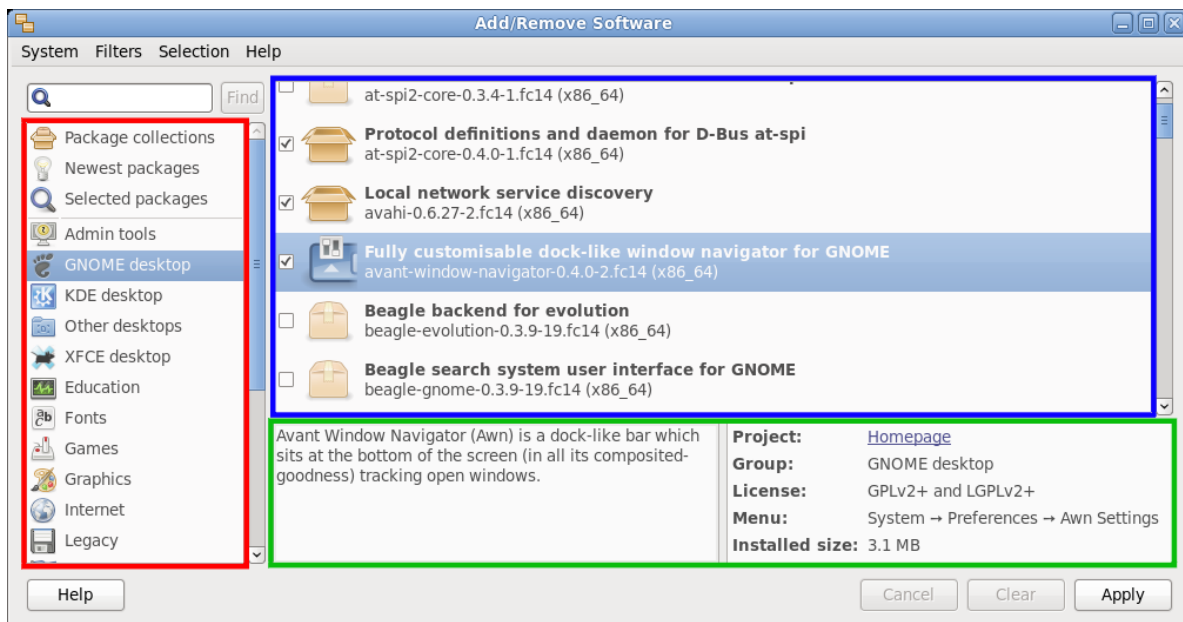


Figure 3.3. gpk-application Listings

3.1.2.1. Package Collections

This list shows the package collections, ie, each entry in the list represents many packages linked in some particular way. If you install a collection, all packages are installed based on that collection.

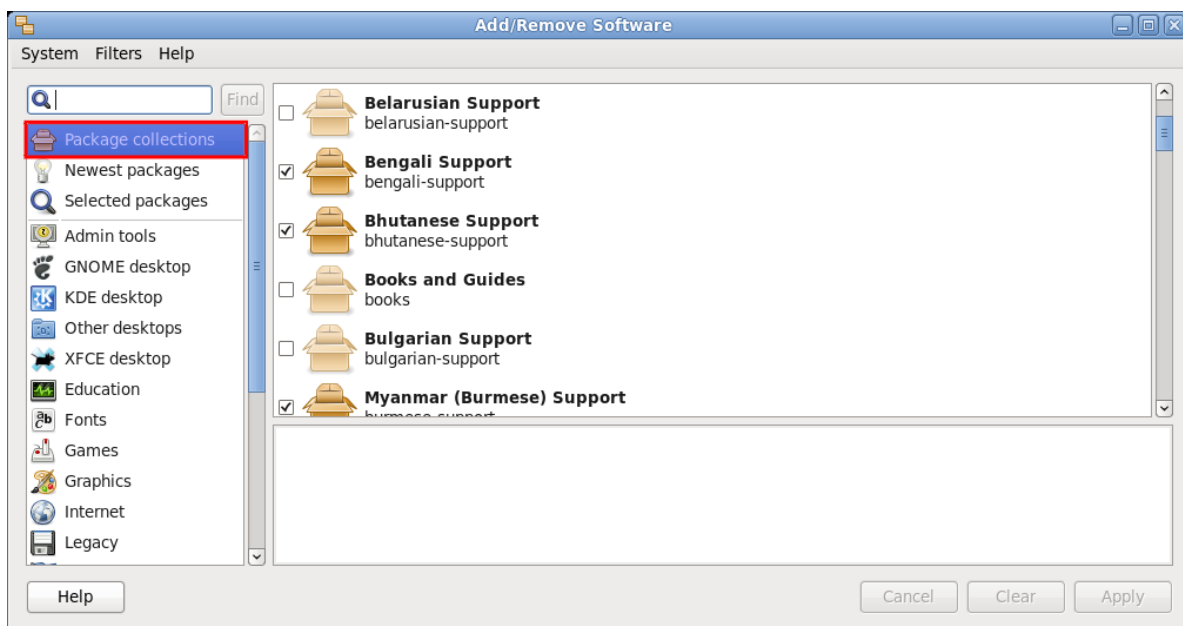


Figure 3.4. gpk-application package collections

3.1.2.2. New Packages

This list shows the latest packages added to the software repositories.

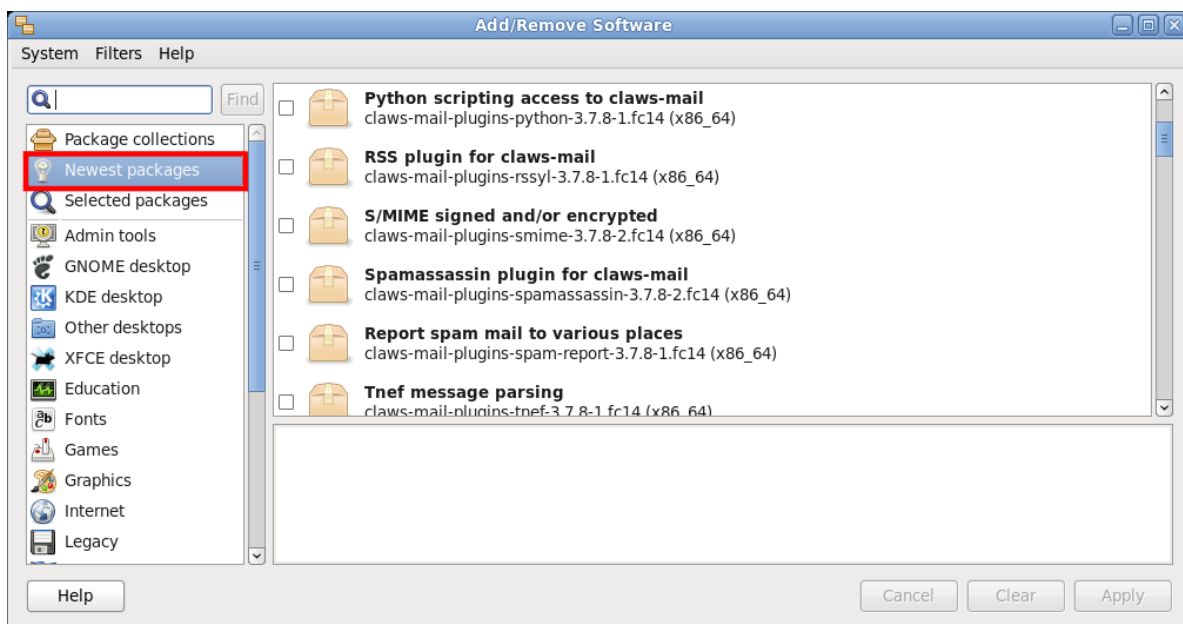


Figure 3.5. gpk-application new packages

3.1.2.3. Popular Collections

Next in the package selection list, after package collections, new packages and selected packages, there is a selected variety of specific collections of popular software. Selecting any of these collections can display all packages that are included in each particular collection.

Below is an example after selecting the list of packages that make up the GNOME Desktop selection.

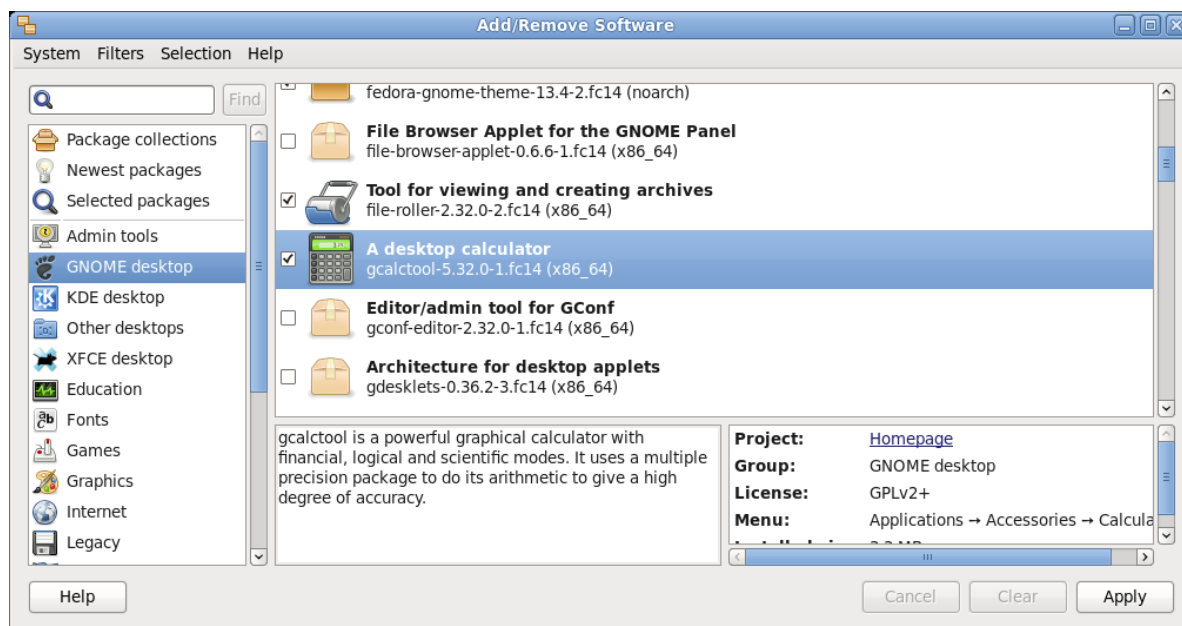


Figure 3.6. gpk-application popular collections

3.1.3. Package Installation

In all gpk-application listings is shown the package or package collection status, checked means installed or to be installed, if not checked then it is not installed or it is scheduled to uninstall, note the change of the icon and the activation of Clean and Apply buttons.

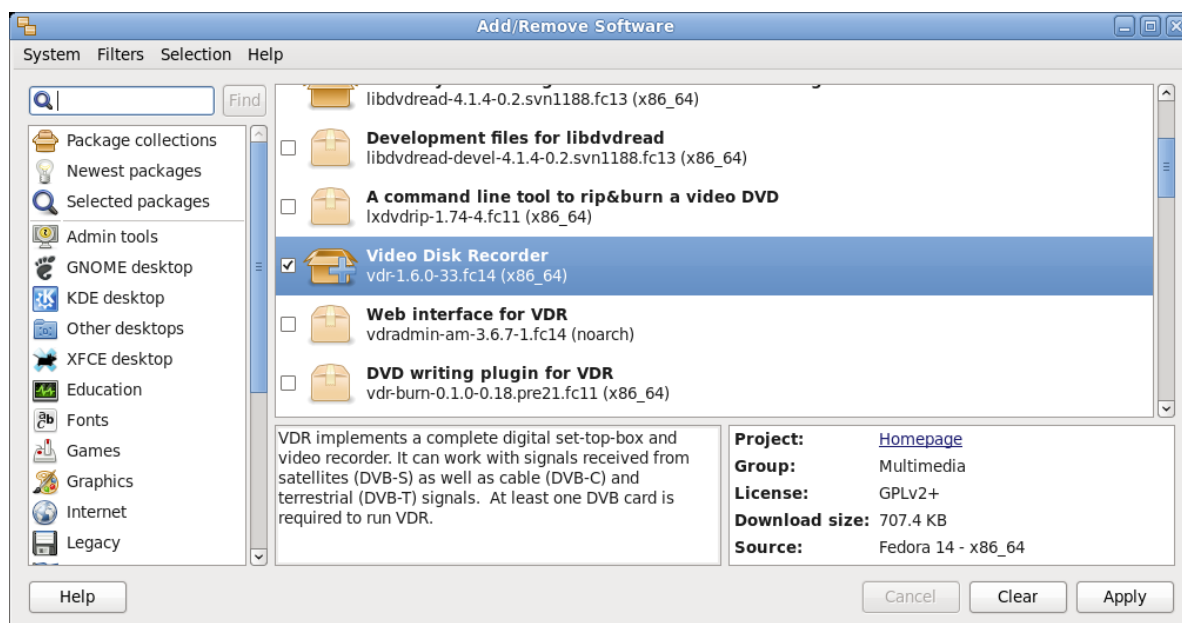


Figure 3.7. gpk-application checked to install

This procedure can be repeated to select multiple packages to install.

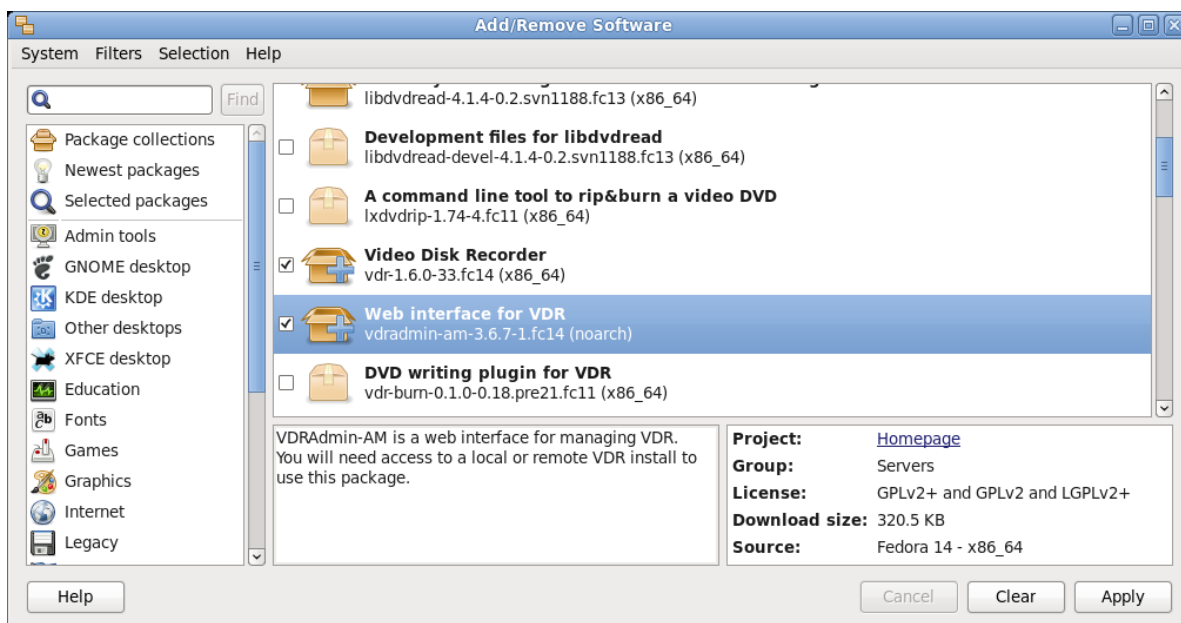


Figure 3.8. gpk-application checked to install

It is possible to review the selected packages by clicking on the entry of the same name in the list of package collections.

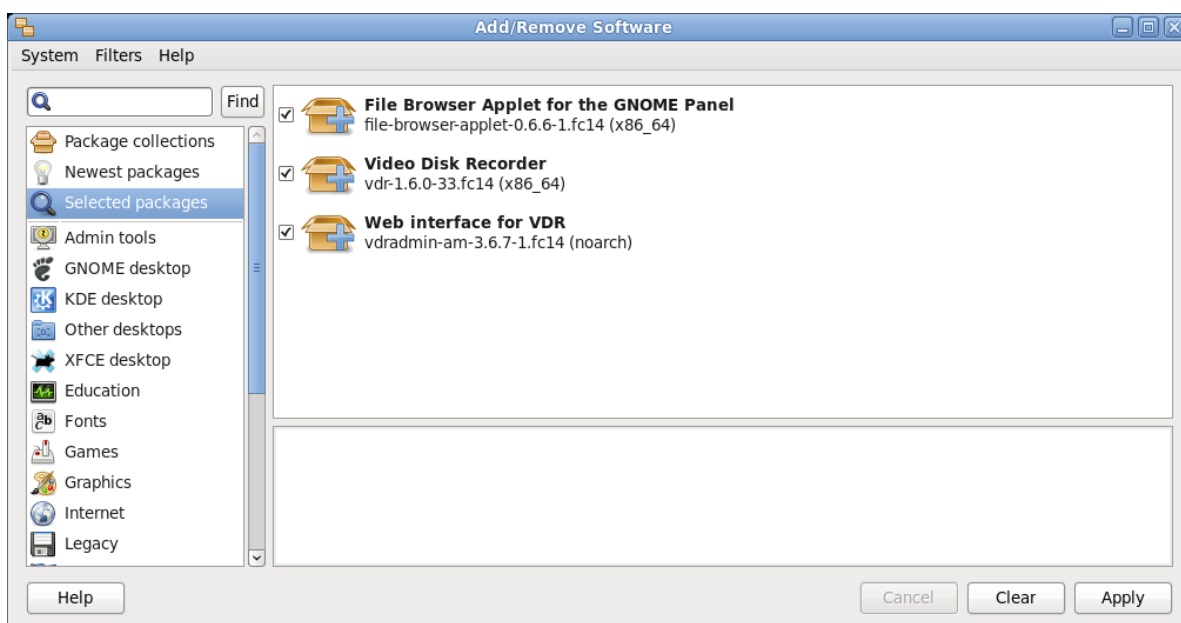


Figure 3.9. gpk-application selected packages

When you finish the package and collection selection, press the Apply button to proceed with scheduled transactions, in this case, to install new software. The system will compute the dependencies and will notify them i agencies if necessary for approval of their installation.

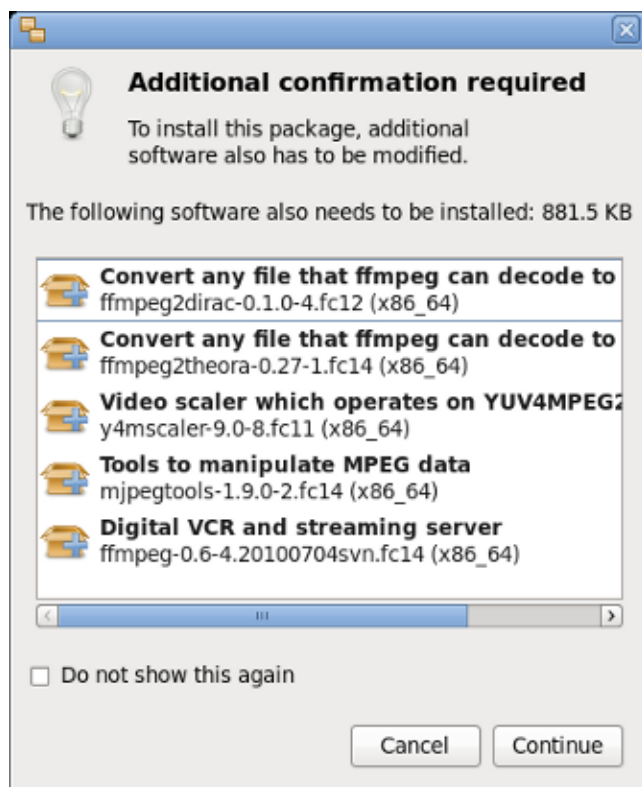


Figure 3.10. gpk-application dependencies to install notification

If you press the Clear button, gpk-application forgets all made selections. In Selected Packages you can customize the selection. It is important to note that gpk-application does not allow to schedule the software removal together with software installation.

If you accept the dependencies installation and press the Install button, the scheduled transactions will go on. As it is an operation that requires root privileges, you must provide the root password before the process can continue.

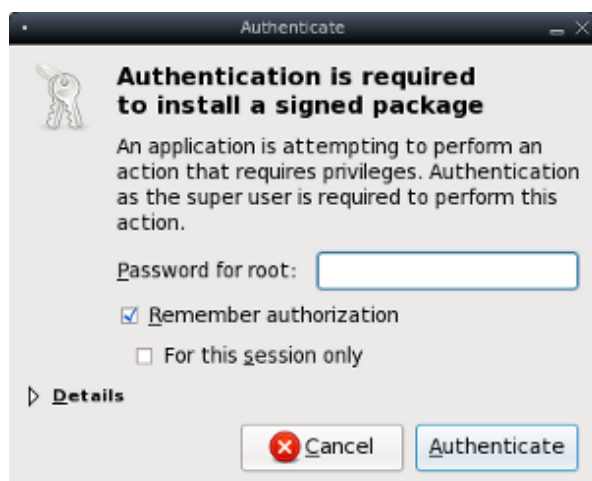


Figure 3.11. gpk-application root authentication

If no such authorization is provided, request it to your system administrator. After authorized, pkg-application will process the scheduled transactions downloading and installing packages.

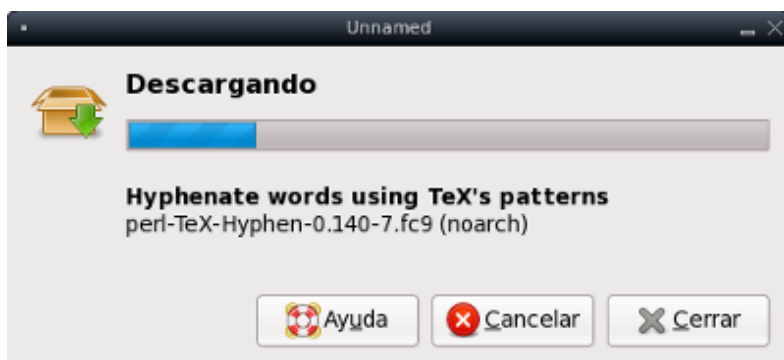


Figure 3.12. gpk-application downloading



Figure 3.13. gpk-application installing

3.1.4. Package Removal

Similar to the installation, package removal with pkg-application is a process of listing, removing the checks to schedule the removal and then to Apply, below are some screenshots.

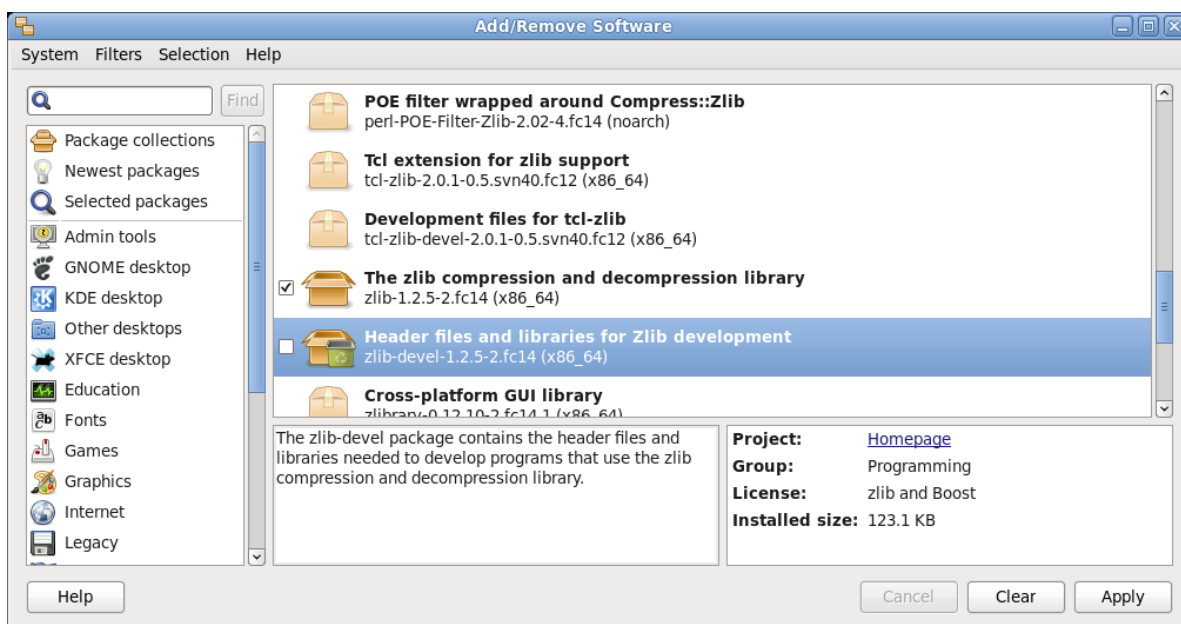


Figure 3.14. gpk-application unchecking to remove

Later, the notification of the dependencies to remove.

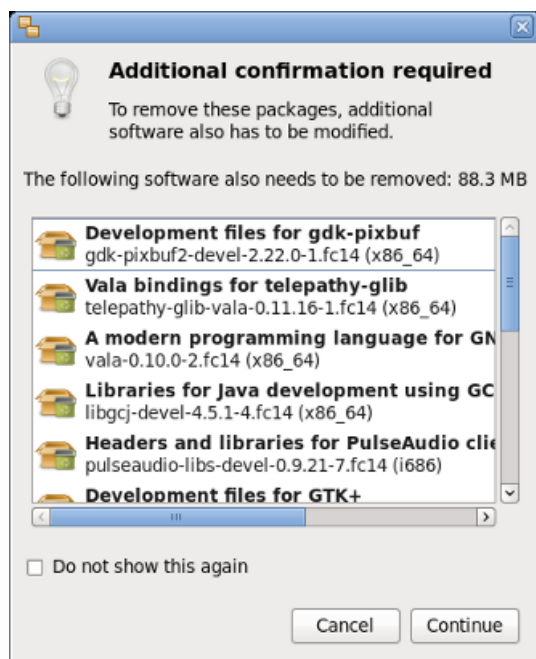


Figure 3.15. gpk-application dependencies removal notification

For brevity we do not show the authentication screen again and we only show the progress of actions.

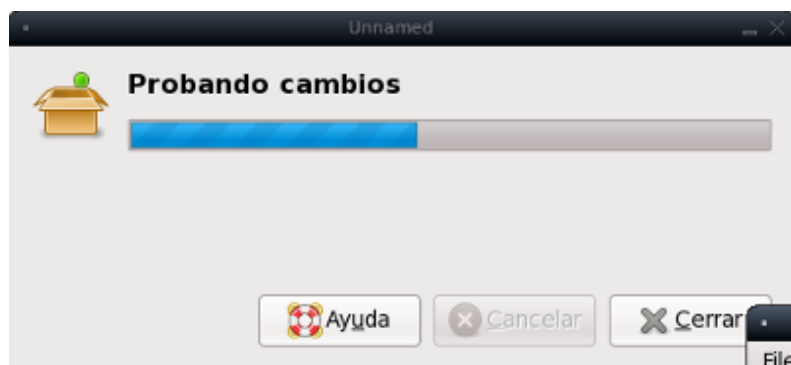


Figure 3.16. gpk-application testing removal

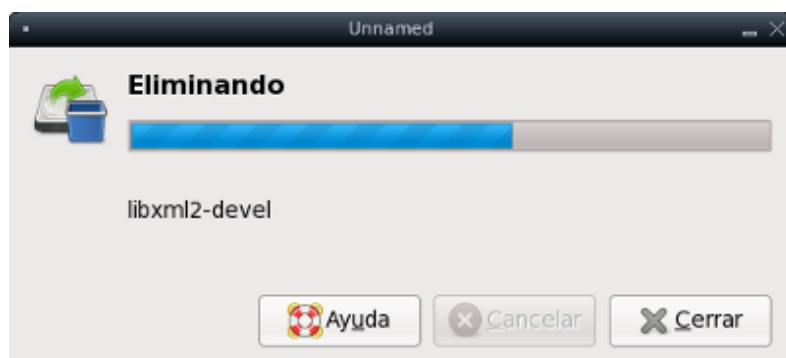


Figure 3.17. gpk-application removing

3.1.5. Listing Filters

To limit the listings it is possible to apply filters and thereby facilitate in the first instance the extent of package exploration.

The first filter is to limit the list of packages to the installed or available.

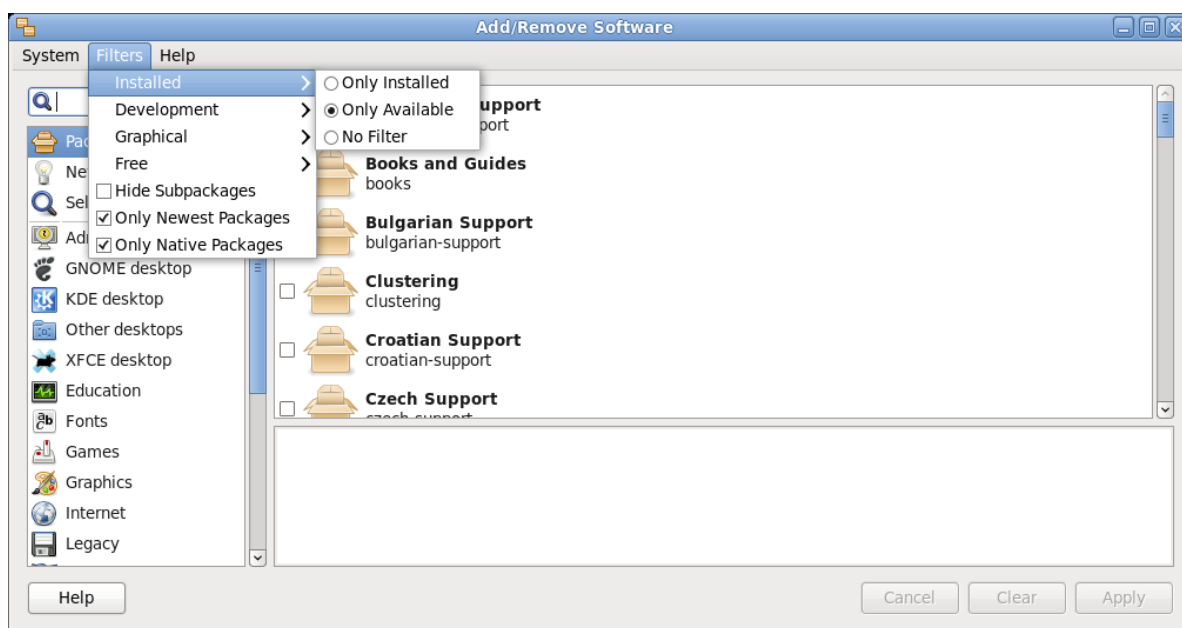


Figure 3.18. gpk-application installed filter

The second filter is to limit the listing to the stable or to the testing packages.

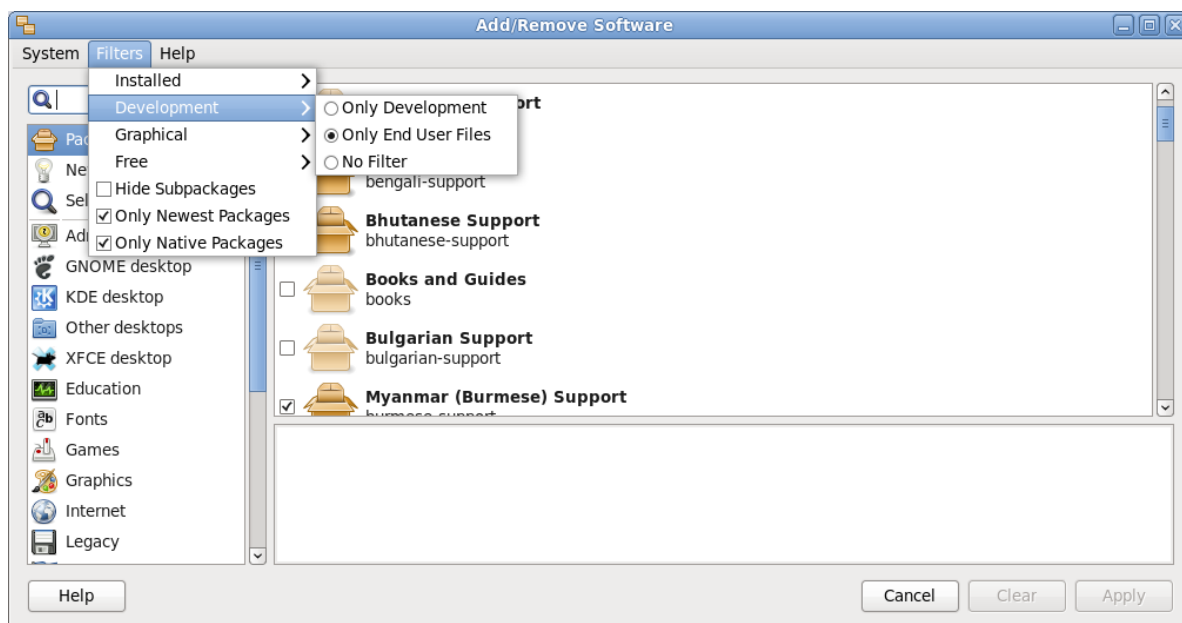


Figure 3.19. gpk-application testing filter.

The third filter is to limit the packages to the GUI (Graphic User Interface) or TUI (Text User Interface) packages. You must have in mind that PackageKit uses heuristic to determine if a package is GUI, and may be wrong if the package has dependencies such as glib2+.

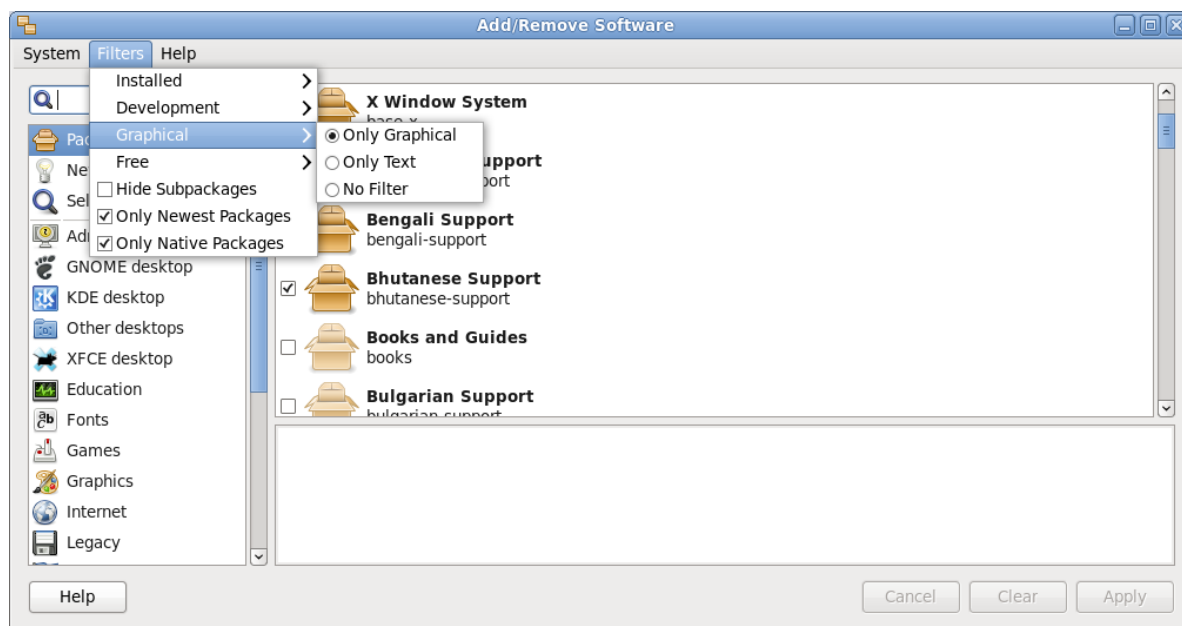


Figure 3.20. gpk-application GUI filter.

The fourth filter is to limit the listing to if the package is free software or not, according to what is considered free software in <http://fedoraproject.org/wiki/Licensing>.

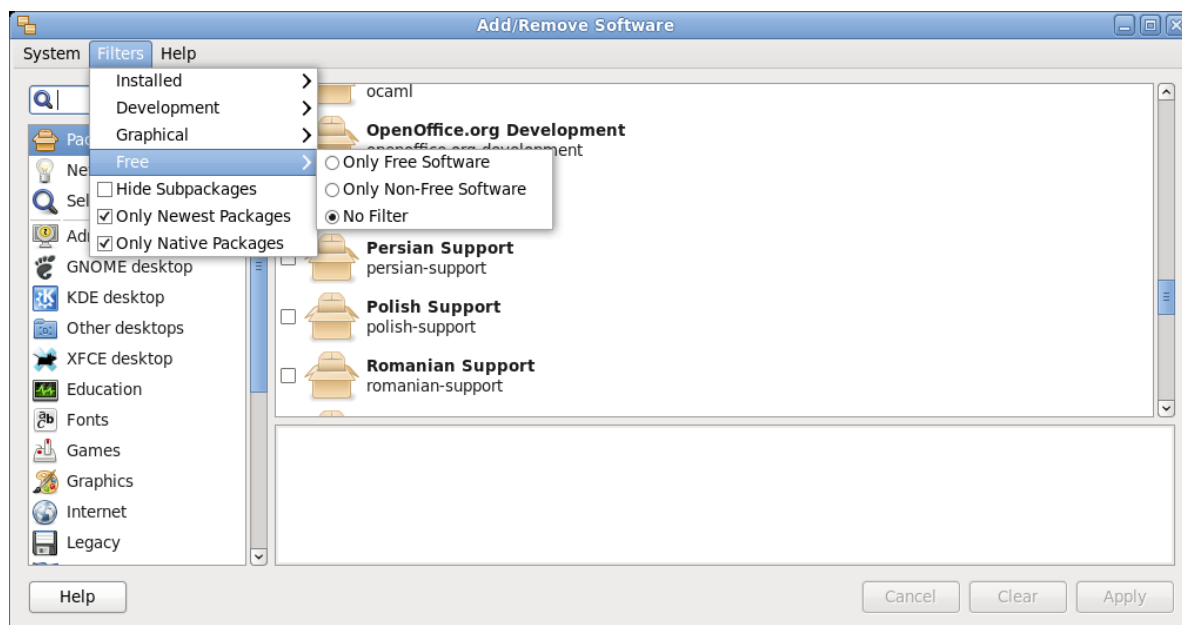


Figure 3.21. gpk-application free filter.

There are other three more options to filter:

Hide subpackages

With this the "children" packages are no listed, only the "parents" are.

New packages only

Will only list the packages recently added to the repositories.

Only native packages

It will only list the packages that matches with the system architecture.

3.2. gpk-log

It is an application to view the software management activity records, below is a screenshot.

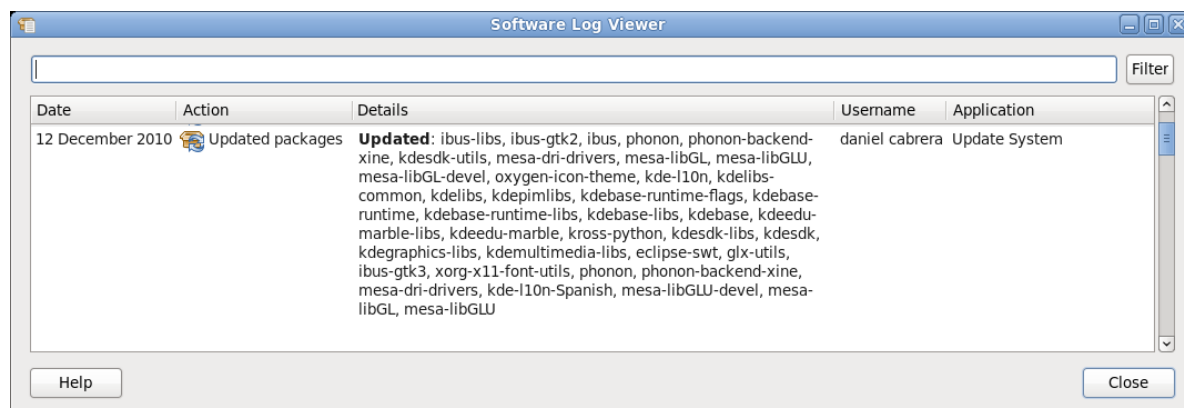


Figure 3.22. gpk-log.

This application can be invoked graphically from gpk-application. Of course gpk-log can be invoked from the shell or any other form of command execution.

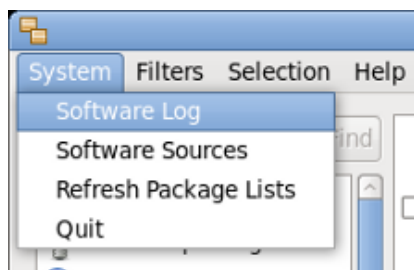


Figure 3.23. gpk-application system menu software log menu item

3.3. gpk-repo

gpk-repo allows to enable and disable the installed software repositories.

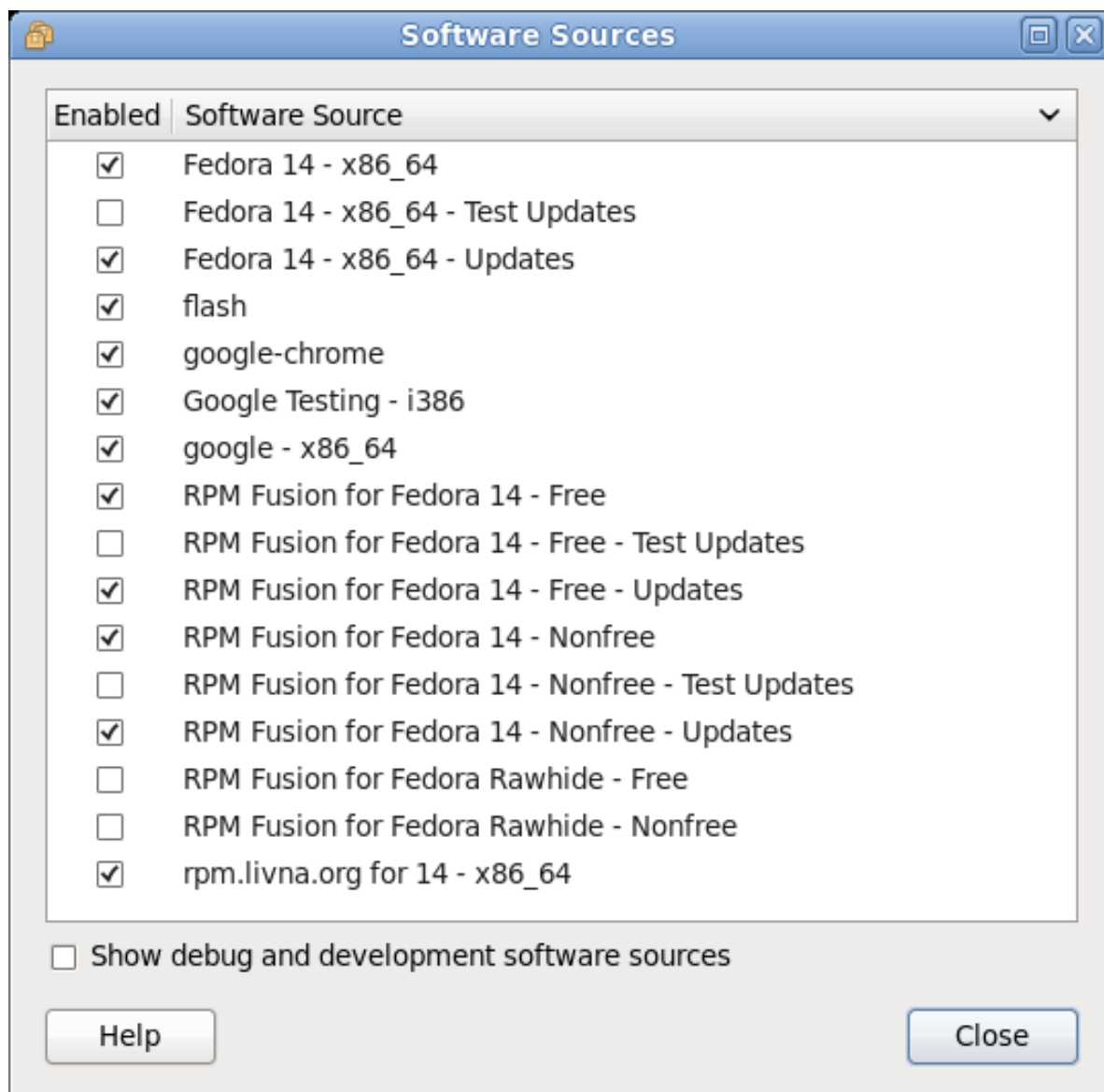


Figure 3.24. gpk-repo

This application can be invoked graphically from gpk-application. Of course gpk-repo can be invoked from the shell or any other form of command execution.

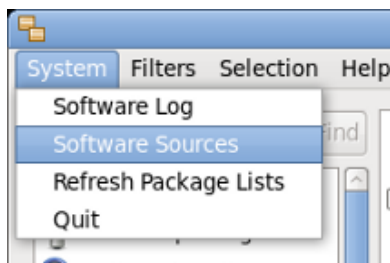


Figure 3.25. gpk-application System->Software Repositories menu option

3.4. gpk-update-viewer

It allows to update the system. It can be invoked from **System** → **Administration** → **Software Update** menu option in GNOME.

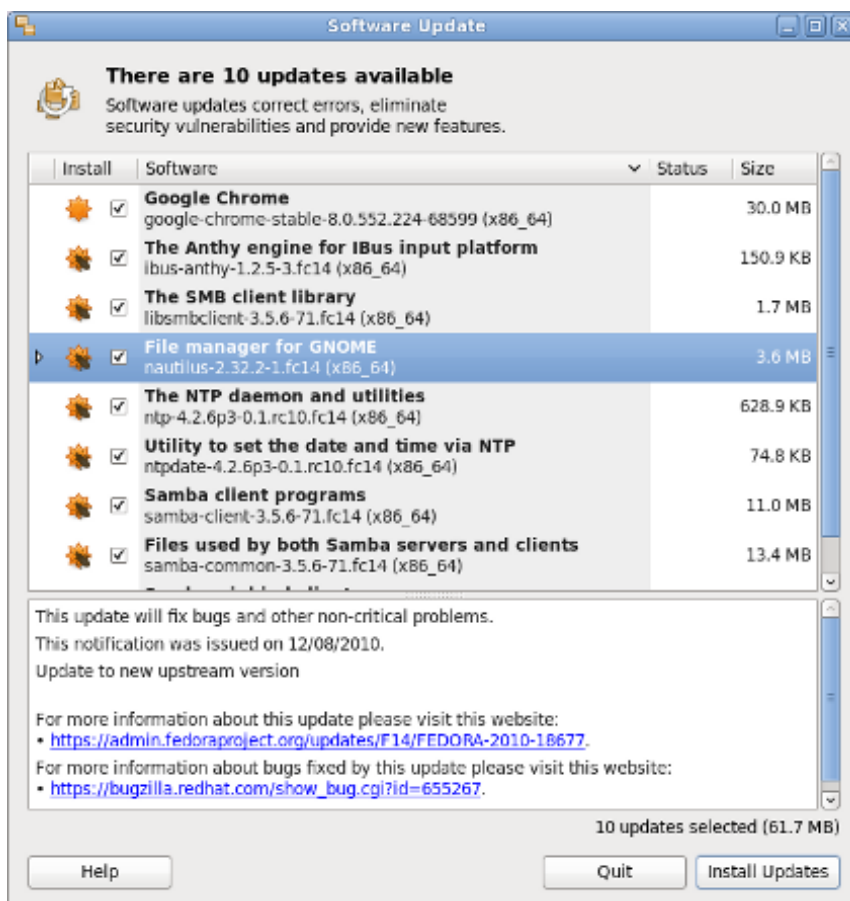


Figure 3.26. gpk-update-viewer

3.5. gpk-update-icon

It is the application that allows to see new updates notification and the PackageKit task queue in the GNOME panel.

When gnome-packagekit is installed, the icon starts automatically when you log into the system. This application displays an icon in the notification area when software updates are available and it is responsible for showing notifications about PackageKit transactions and events. It is also responsible to invoke gpk-update-viewer for system updates.



Figure 3.27. gpk-update-icon

3.6. gpk-prefs

It configures the updates preferences. It can be invoked from **System** → **Preferences** → **Software Updates** menu option in GNOME.

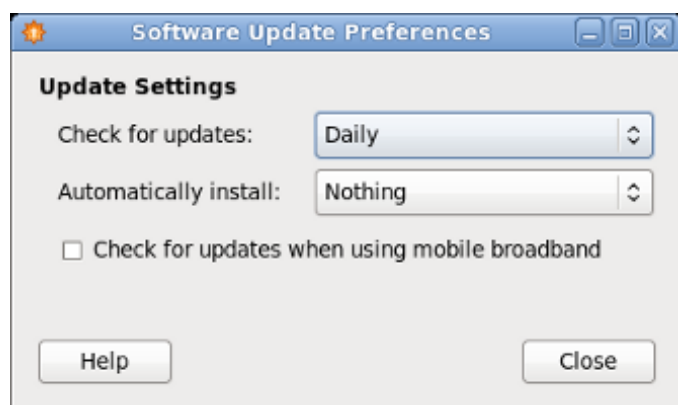


Figure 3.28. `gpk-prefs`

YUM Configuration

This section of the manual will review the YUM system configuration to be able to understand its overall functionality.

The general configuration of the YUM system is found in **/etc/yum.conf** and the configured YUM repositories are defined in **/etc/yum.repos.d/**.

To understand the configurations we will analyze the configuration files once Fedora is installed. For more advanced configurations, one can always consult the online manual pages **man yum** and **man yum.conf**.

4.1. General Configuration

To begin, start your favorite text editor and open the general configuration file for YUM, **/etc/yum.conf**, and you will see content similar to what is shown below.

/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
installonly_limit=2

# This is the default, if you make this bigger yum won't see if the metadata
# is newer on the remote and so you'll "gain" the bandwidth of not having to
# download the new metadata and "pay" for it by yum not having correct
# information.
# It is esp. important, to have correct metadata, for distributions like
# Fedora which don't keep old packages around. If you don't like this checking
# interrupting your command line usage, it's much better to have something
# manually check the metadata once an hour (yum-updatesd will do this).
# metadata_expire=90m

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

The first notable aspect of the YUM configuration file is the block structure with sections identified with brackets: *[section]*. This structure repeats when the repositories are configured. In this case **yum.conf** in the section *[main]* define the general behavior of the YUM system, and establish the base directives, for example, where to store the cache. Each of the directives mentioned are commented on below.

cachedir

Directory where YUM should store its cache and file db. The default value is **/var/cache/yum**.

keepcache

Can be '1' or '0'. It determines whether YUM should keep a cache of the headers and packages after a successful installation. Default value is '1' (keep the files).

debuglevel

Level of

logfile

Complete path to the name of the file where YUM should write its logging.

exactarch

Can be '1' or '0'. Setting it to '1' causes, for example, **yum update** to only update the packages of the architecture that is installed. With the option active, the command **yum** won't install an i686 package to update an i386 package. The default value is '1'.

obsoletes

This option only has an effect during an upgrade. It enables the YUM obsolescence processing logic. It is useful when a distribution level upgrade is being done. See also: the documentation of the command **yum upgrade** for more details. Command line option: **--obsoletes**.

gpgcheck

Can be '1' or '0'. This indicates to YUM whether or not it should check the GPG signature of the packages. When it is enabled in *[main]* section, it establishes a default behavior for all of the repositories. This option also determines whether an installation of a local RPM will have its GPG signature checked or not. The default value is '0'.

plugins

Either '0' or '1'. It is a global switch to enable or disable YUM extensions (plugins). Its default value is '0' (extensions disabled). See the PLUGINS section of yum man page (man 8 yum) for more information about installing YUM extensions.

metadata_expire

Duration in seconds after which the metadata will expire. So if the current metadata downloaded is some seconds younger, then **yum** will not update the repository metadata. If you think that **yum** is not downloading the update information as often as you wish, reduce the value of this parameter. You can also change the default format in seconds to days, hours or minutes adding *d*, *h* or *m* to the specified value. The default is 1.5 hours, for running **yum-updatesd** every hour. You can also use the word *never* meaning that the metadata will never expire.

Due to the nature of certain software, in particular the kernel, it is necessary to have installed more than one version at a time for different reasons, the following parameters are relevant in such cases.

installonlypkgs

List of packages that must be always installed, never updated. In particular the **kernel** falls into this category. The default value is **kernel**, **kernel-smp**, **kernel-bigmem**, **kernel-enterprise**, **kernel-debug**, **kernel-unsupported**.

installonly_limit

Number of packages listed in *installonlypkgs* to be kept installed at the same time. Setting this value to 0 disables the function. The default value is '2'.

These are just some YUM system configuration directives, for more, **man yum.conf**. You may include the repositories configuration within **yum.conf** but that kind of configuration is deprecated and it is preferred to use individual files in **/etc/yum.repos.d/** which will be described later in this manual.

4.2. YUM Variables

Several YUM variables are available for use in the configuration files of the YUM system. They can be used with various configuration options, such as: *name*, *baseurl* as well as with the commands.

\$releasever

It will be replaced with the value of the version of the package specified by *distroverpkg*. The default is the version of redhat-release package. In Fedora it is *fedora-release*.

\$arch

It is replaced with the architecture of your system and according to `os.uname()` in Python.

\$basearch

Is replaced by the base architecture in YUM. For example, if your **\$arch** is **i686**, **\$basearch** would be **i386**.

\$uuid

Will be replaced by a unique, persistent uuid for this machine. The generated value will be stored in **/var/lib/yum/uuid** and will be reused until this file is deleted.

\$YUM0-\$YUM9

These variables will be replaced with the value of the shell variable of the same name. If the shell environment variable does not exist, then the variable in the configuration file will not be replaced.

Since yum version 3.2.28 every file in **/etc/yum/vars** becomes a variable with the same name as the file name and overwrite any of the variables previously described above.

Make sure the file in **/etc/yum/vars** are readable because there will be no warnings or errors otherwise. From the files it will only be read the first line and the new line character will be ignored. Note that there is no check for the existence of strange characters.

4.3. Cache Management

By default the current versions of yum erase the data and packages that they have downloaded immediately after they have been successfully used in an operation by yum. This minimizes the storage space required to operate. However, you can enable a cache system to cause yum to retain the files it has downloaded. The advantages are the following:

1. Improve on the performance.
2. You can operate yum without an internet connection.
3. You can copy packages from the cache and reuse them at will.

By default, yum stores its temporary files in **/var/cache/yum/** with a subdirectory for the configured repository. The directory **packages/** inside of each subdirectory maintains the packages in the cache. For example, **/var/cache/yum/x86_64/13/fedora/packages** maintains the downloaded packages of the fedora repository.

To enable the use of the cache, use the directive *keepcache* in **/etc/yum.conf**.

/etc/yum.conf

```
[main]
cachedir=/var/cache/yum
keepcache=1
gpgcheck=1
plugins=1
...
```

To clear the cache you must use the **yum clean** command with some of its options:

yum clean expire-cache

Removes the local data if necessary for each repository. This means that yum will revalidate the cache for each repo, if the cache is invalid nothing will be removed.

yum clean packages

Delete the packages from the cache.

yum clean headers

Delete the package headers from the cache.

yum clean metadata

Removes all files that yum uses to determine the package availability. This command will force yum to download all the necessary metadata the next time you use it.

yum clean dbcache

Removes the sqlite cache used for faster access to metadata. This command will force the download of yum metadata the next time it runs.

yum clean rpmdb

Removes the data from local rpmdb cache.

yum clean plugins

Causes the enabled plugins to delete their cached data.

yum clean all

Execute all of the previous commands.

4.4. Priority Configuration

Following with yum configuration, we will show how to configure Fedora to make yum use the repositories on priority based order without discarding the external mirrors list. Let us see the needed configuration.

```
[fedora]
name=Fedora $releasever - $basearch
failovermethod=priority
baseurl=http://mirepo.org/$releasever/$basearch/os/
        http://otrorepo.org/$releasever/$basearch/os/
mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=fedora-$releasever&arch=$basearch
enabled=1
metadata_expire=7d
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora-$basearch
```

First note the use of multiple URLs in *baseurl*. It should be done as set here, and do not use multiple *baseurl* statements cause it will not work properly.

Then note the use of the *failovermethod* statement, this parameter can have values 'roundrobin' or 'priority'. The default value is 'roundrobin' that randomly selects a URL from the list of URLs. When the value is 'priority' it means to start from the first URL from the list of URLs. It is important to note here that the list of URLs is constructed from the *baseurl* statement and then from the list obtained from *mirrorlist*. In that way the desired goal is achieved, searching in the local repositories first, and then, if they fail, in the specified order, searching in the URLs from the mirrors list.

4.5. Bandwidth Control

In managed networks of workstations or servers there is a need to control the use of the available bandwidth for every YUM system, and it is also useful for users who want to keep the YUM system under control particularly when they are doing something else that requires bandwidth.

The solution is simple, it is to configure yum to control itself the use of bandwidth, below is an example of a **yum.conf** with the relevant parameters. *throttle* can be an absolute value or a percentage value, in the latter case, *bandwidth* specifies 100% of available bandwidth in bytes/sec.

If you wish to disable the control, put *throttle=0*.

```
# Adjustments for bandwidth control
# 70% of 512kbps ~ 358Kbps limit
throttle=70%
bandwidth=64000
```

4.6. Proxy Configuration

In managed networks where there is an HTTP proxy that regulates which workstations and servers can access internal or external networks with for example HTTP protocol, yum must be configured appropriately in order to use the proxy resources to be able to access to software repositories when needed.

In **yum.conf** you can use the following three directives:

proxy

URL of proxy server that yum should use.

proxy_username

Username to use with the proxy server.

proxy_password

Password for this proxy server.

Remember that YUM may use available repositories via HTTP and FTP so it is not necessarily an HTTP proxy server.

Relevant excerpts from a **/etc/yum.conf** example configuration:

```
proxy=http://proxy.domain.com:3128
proxy_username=dragon
proxy_password=secret
```

4.7. Multiarchitecture Policy

For the system to support multi-arch packages, such as x86_64, you can install one or both versions of a given package. To determine what to do in general you should define the *multilib_policy* configuration parameter.

multilib_policy all|best

all means that yum will install all possible architectures of any package you wish to install. Then **yum install foo** will install foo.i386 and foo.x86_64 in x86_64 systems, if available. *best* means installing only the best architecture for this platform.

Relevant excerpts from a **/etc/yum.conf** example configuration:

```
multilib_policy best
```

4.8. Management of YUM Repositories

When you install a Fedora system, files that declare the base repositories are already installed and the base repositories enabled and available, in fact, during the installation process **Anaconda** must have given you the opportunity to enable some other repositories. In this section we will review the ways to add new repositories and how to adjust their settings in a particular way to suit the needs of the administrator.

4.8.1. Add New Repositories

Two methods are used to add new repositories, most suitable for the systems administrator is the provision of the files that make up the repository via an installable rpm file or by supplying the .repo configuration simple text file from the repository owners.

Example with a rpm file that configures the desired repository:

```
$ su -
<password>
# rpm -vhi http://download1.rpmfusion.org/free/fedora/rpmfusion-free-release-
stable.noarch.rpm
```

Example with a simple .repo text file:

```
$ su -
<password>
# cd /etc/yum.repos.d/
# wget http://repos.fedorapeople.org/repos/mmcgrath/nagios/fedora-nagios.repo
....
Saving to: "fedora-nagios.repo"

100%[=====>] 220          --.-K/s   in 0s

2010-07-30 18:30:14 (24.9 MB/s) - "fedora-nagios.repo" saved [220/220]
```

Certainly it is always possible to launch your favorite text editor and start a manual configuration.

4.8.2. Manual Configuration of a Repository

Next, we will review as an example the manual configuration of a particular .repo file to examine its details and to take control of each one of them.

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

```
[fedora]
name=Fedora $releasever - $basearch
failovermethod=priority
baseurl=http://download.fedoraproject.org/pub/fedora/linux/releases/$releasever/Everything/
$basearch/os/
#mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=fedora-$releasever&arch=$basearch
enabled=1
metadata_expire=7d
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora-$basearch

[fedora-debuginfo]
```

```

name=Fedora $releasever - $basearch - Debug
failovermethod=priority
#baseurl=http://download.fedoraproject.org/pub/fedora/linux/releases/$releasever/Everything/
$basearch/debug/
mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=fedora-debug-$releasever&arch=
$basearch
enabled=0
metadata_expire=7d
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora-$basearch

[fedora-source]
name=Fedora $releasever - Source
failovermethod=priority
#baseurl=http://download.fedoraproject.org/pub/fedora/linux/releases/$releasever/Everything/
source/SRPMS/
mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=fedora-source-$releasever&arch=
$basearch
enabled=0
metadata_expire=7d
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora-$basearch

```

As in many Linux configuration files, lines beginning with # are comments and they are ignored by yum.

Each *[nombre]* section defines a yum repository, then within each repository definition you can overwrite the general behavior of yum with global scope policy for handling this particular repository or you can add new specific directives at repository scope.

```

[fedora] << repository identifier
name=Fedora $releasever - $basearch
baseurl=http://download.fedoraproject.org/pub/fedora/linux/releases/$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

```

The repository identifier must be unique.

name

A string (name) of characters that describes the repository for the human.

baseurl

Must be a URL that points to the directory where the yum 'repodata' is. It can be a http://, ftp:// or file:// URL. You can specify multiple URLs in a statement, the best way to do it is:

```

[repositoryid]
name=Some name for this repository
baseurl=url://server1/path/to/repository/
        url://server2/path/to/repository/
        url://server3/path/to/repository/

```

You can use HTTP basic authentication prefixing "usuario:contraseña@" to the server name in the line *baseurl*, for example *baseurl=http://usuario@passwd@example.com/*.

mirrorlist

It specifies a URL that points to a file containing a list of baseurls. You can use this statement instead of *baseurl*.

enabled

It can be '1' or '0'. This tells yum to whether or not use this repository.

gpgcheck

It can be '1' or '0'. This tells yum whether or not to perform a GPG signature verification on packages obtained from this repository.

gpgkey

A URL that points to a file containing ASCII GPG key for the repository. This option is used if yum needs a public key to verify the packages and the required key has not been imported into the RPM database. If this option is placed yum will automatically import the key specified by the URL. yum will ask for confirmation before installing the key unless you specify *assumeyes*. Also you can list multiple URLs in the same way as it was done for *baseurl*.

failovermethod

It can be 'roundrobin' or 'priority'. 'roundrobin' randomly selects a URL from the list of URLs to start and then if it fails to contact the selected server it proceed with another selected similarly. 'priority' starts from the first one listed in *baseurl* and if it fails to contact the server, it continues sequentially with the next in the list. If not specified, the default is 'priority'.

4.8.3. Removing Repositories

To remove the xxx repository you should remove its .repo configuration files from **/etc/yum.repos.d/**. However, each .repo file can and usually have more than one yum repository defined in it, but it is usual that they are all interrelated in a "binary", "sources" and "debug-info", ie. .repo includes the configuration of three interlinked yum repositories. If so just delete the .repo file e.g. rpmfusion-free.repo, otherwise, be careful to just remove the appropriate sections of the repositories you wish to remove, using a text editor of your preference.

/etc/yum.repos.d/rpmfusion-free.repo

```
[rpmfusion-free]
name=RPM Fusion for Fedora $releasever - Free
baseurl=http://download1.rpmfusion.org/free/fedora/releases/$releasever/Everything/$basearch/
os/
#mirrorlist=http://mirrors.rpmfusion.org/mirrorlist?repo=free-fedora-$releasever&arch=
$basearch
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-rpmfusion-free-fedora

[rpmfusion-free-debuginfo]
name=RPM Fusion for Fedora $releasever - Free - Debug
#baseurl=http://download1.rpmfusion.org/free/fedora/releases/$releasever/Everything/
$basearch/debug/
mirrorlist=http://mirrors.rpmfusion.org/mirrorlist?repo=free-fedora-debug-$releasever&arch=
$basearch
enabled=0
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-rpmfusion-free-fedora

[rpmfusion-free-source]
name=RPM Fusion for Fedora $releasever - Free - Source
#baseurl=http://download1.rpmfusion.org/free/fedora/releases/$releasever/Everything/source/
SRPMS/
mirrorlist=http://mirrors.rpmfusion.org/mirrorlist?repo=free-fedora-source-$releasever&arch=
$basearch
enabled=0
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-rpmfusion-free-fedora
```


4.8.4. The Fedora Repositories

Fedora repositories are included in the `fedora-release` package. The actual `.repo` files included are:

```
/etc/yum.repos.d/fedora.repo  
/etc/yum.repos.d/fedora-updates.repo  
/etc/yum.repos.d/fedora-updates-testing.repo
```

fedora.repo defines the following three repositories:

- `[fedora]`
- `[fedora-debuginfo]`
- `[fedora-source]`

fedora-updates.repo defines the following three repositories:

- `[updates]`
- `[updates-debuginfo]`
- `[updates-source]`

fedora-updates-testing.repo defines the following three repositories:

- `[updates-testing]`
- `[updates-testing-debuginfo]`
- `[updates-testing-source]`

From these nine repositories, by default are enabled:

- `[fedora]`
- `[updates]`

The rest are disabled and they are used by developers, programmers and testers to improve existing programs in these repositories.

In particular it is interesting that you enable `[updates-testing]` if you wish to support the task of testing new programs being published in Fedora. The `[updates-testing]` repository is a precursor to the repository `[updates]` where the packages are considered stable and are declared so to alert users that they may have minor problems. If you have a Fedora FAS account, then you can vote and give a positive or negative karma according to your experience with the new program under testing in [bodhi](https://admin.fedoraproject.org/bodhi)¹.

For Fedora developers there is an additional package that adds another `.repo` file with Fedora repositories, `/etc/yum.repos.d/fedora-rawhide.repo`, also known as *rawhide*.

```
$ su -c "yum install fedora-release-rawhide"
```

Install *fedora-release-rawhide* only if you know what you are doing, never leave this repository enabled by default if you are newbie, in this repository is where Fedora development occurs and there are

¹ <https://admin.fedoraproject.org/updates>

versions that are not even guaranteed to work. By default all rawhide repositories are disabled when you install the package `fedora-release-rawhide`.

`fedora-rawhide.repo` defines the following three repositories:

- `[rawhide]`
- `[rawhide-debuginfo]`
- `[rawhide-source]`

4.8.5. Third Party and Personal Repositories

In the Fedora history there have been many external yum repositories provided by third parties to supply those packages that were not in Fedora for one reason or another. In many cases, these repositories have been helpful, and in many others they have not, and have produced much confusion in the regular user.

The quality of these repositories is not tested, guaranteed or endorsed in any way by the Fedora Project. If you decide to use any of these third-party repositories or personal, you are on your own.

Fedora Project generally discourages the use of such third-party repositories as possible. Besides, having this policy of using the least number of repositories as possible, it will increase the chance to not suffer problems when upgrading your system from a Fedora version to another.

As for the availability or not of certain software in the Fedora repositories, we encourage you to directly contribute directly to the Fedora Project if you meet the relevant guidelines.

When Fedora Project is developing a new version, it does not take care of the external third party or personal repositories and only provides supports for the packages formally provided in the official repositories.

In short, there may be third party repositories with excellent quality and compatibility with Fedora, and others not and that will only create problems. Use them with caution, at your own risk and enable such repository only for the moment and for specific issues.

```
$ su -c "yum install --enablerepo=drivers nvidia-driver"
```

4.8.6. List of Mirrors and Metalink

The *mirrorlist* and *metalink* configuration options are linked in Fedora yum and this section will provide an overview of their operations in each case and their relationship.

The repository configuration of a Fedora box contains the following *mirrorlist* directive:

```
...
mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=fedora-$releasever&
arch=$basearch
...
```

Using the replacement of corresponding variables, *\$releasever=f13* and *\$basearch=x86_64*, for example, and using a web browser to download, you can get the following xml document.

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<metalink version="3.0" xmlns="http://www.metalinker.org/" type="dynamic" pubdate="Sat,
21 Aug 2010 14:10:47 GMT" generator="mirrormanager" xmlns:mm0="http://fedorahosted.org/
mirrormanager">
  <files>
    <file name="repomd.xml">
      <mm0:timestamp>1274247846</mm0:timestamp>
      <size>4400</size>
      <verification>
        <hash type="md5">fa0f764c57ba5b15be4e188efe9d8494</hash>
        <hash type="sha1">c83b44fb1ccf030b7b81ae9bf34572d714ab4e1a</hash>
        <hash
type="sha256">7b00cc2cf43ff34baedcc16c389a015f2aad5e35f725708b5afef56bdb1e13f6</hash>
        <hash
type="sha512">2d1f5bcde26213d9ead4b5847501f9315d9391b73eabdd56dd1b45e6307ccd46b4d218c2bcf750364449fba25
hash>
      </verification>
      <resources maxconnections="1">
        <url protocol="http" type="http" location="BR" preference="100" >http://
fedora.c3sl.ufpr.br/linux/releases/13/Everything/x86_64/os/repodata/repomd.xml</url>
        <url protocol="ftp" type="ftp" location="BR" preference="100" >ftp://
fedora.c3sl.ufpr.br/fedora-enchilada/linux/releases/13/Everything/x86_64/os/repodata/
repomd.xml</url>
        <url protocol="rsync" type="rsync" location="BR" preference="100" >rsync://
fedora.c3sl.ufpr.br/fedora/linux/releases/13/Everything/x86_64/os/repodata/repomd.xml</url>
        <url protocol="http" type="http" location="BR" preference="99" >http://
mirror.ispbrasil.com.br/fedora/releases/13/Everything/x86_64/os/repodata/repomd.xml</url>
        <url protocol="http" type="http" location="BR" preference="98" >http://
fedora.pop.com.br/linux/releases/13/Everything/x86_64/os/repodata/repomd.xml</url>
        <url protocol="http" type="http" location="CR" preference="97" >http://
mirrors.ucr.ac.cr/fedora/releases/13/Everything/x86_64/os/repodata/repomd.xml</url>
      </resources>
    </file>
  </files>
</metalink>
```

You may see here that it is not simple list of baseurls. The definitions of both settings are explained below.

mirrorlist

Specifies a URL to a file containing a baseurls list. It may be used as a replacement o as complementing the *baseurl* option. Variables are allowed with this option.

metalink

Specifies a URL to a *metalink* file that point to a **repomd.xml** file, it generates a list of mirrors for the repository converting the mirrors in the **repomd.xml** file to a *baseurl*. The metalink file also contains the timestamp of the data in the repomd.xml file, its length and checksum. Data are checked versus the repomd.xml file file and all information downloaded from metalink file must match. *metalink* can be used as a substitute or complement to *baseurl*. Variables can be used with this directive. This option disables the *mirrorlist* option.

But also in both sections of the documentation a special hack is mention and says to use the word "metalink" in *mirrorlist* option, and then the value of *mirrorlist* is copied to *metalink* (if metalink is not defined). Fedora uses this hack by default, and this is their relationship.

In general, the out of the box configuration of Fedora should offer the best nearby available family of repositories to the user's installation, note that above all the list provided by Brazil and the user is from Venezuela, and in Venezuela there isn't a Fedora mirror. In Fedora's **mirrormanager**, geographical information is used to list appropriate mirrors.

Nevertheless, for example, Brazil is not as well connected to Venezuela as Venezuela is connected to the US. As such, the Venezuelans generally prefer to use US mirrors to improve their download speed. If the **plugin yum-fastest-mirror** is used, it will only use the list of calculated baseurls to

choose the repository, which will never contain more than just the Brazilian repositories. Solution: hack and reconfigure *mirrorlist* and/or *metalink*.

One option is to simply put a list specifying multiple *baseurl*, but what are they?. Is there a way to automate? Of course it is also important that repositories are synchronized with the sources.

- <https://fedorahosted.org/mirrormanager/>
- <http://www.metalinker.org/>

mirrorlist_expire

Specifies the time in seconds after which the local list of mirrored cache expires. If the local list is older than this time, **yum** will download a new list.

4.8.7. Configuration for Package Protection

This feature was recently added to **yum** from a plugin called **yum-plugin-protect-packages** and prevents **yum** from uninstalling itself and prevents to remove other protected packages. **yum** is by default the only protected package, but by extension this automatically protects everything that **yum** depends on (rpm, python, glibc, etc.). So then this functionality is very good even without the need to carefully construct lists of important packages.

In a standard way it is blocked the uninstall of **yum** and its dependencies, but if you want to add and protect other packages, you must create files in **/etc/yum/protected.d/*.conf** and add a package name per line in each one of them.

Example situation, when you wish to protect the *postgresql* package. Configure a .conf file either creating or editing it, for example **rdbsms.conf** in the appropriate directory.

```
$ vi /etc/yum/protected.d/rdbsms.conf
postgresql
:wq
```

Test to verify that YUM cannot uninstall the configured package by error.

```
# yum remove postgresql
Loaded plugins: aliases, changelog, fastestmirror, filter-data, keys, list-data, presto,
refresh-packagekit, verify
Setting up Remove Process
Resolving Dependencies
--> Running transaction check
---> Package postgresql.i686 0:8.4.4-1.fc13 set to be erased
--> Processing dependencies: postgresql = 8.4.4-1.fc13 for package: postgresql-
devel-8.4.4-1.fc13.i686
--> Processing dependencies: postgresql = 8.4.4-1.fc13 for package: postgresql-
server-8.4.4-1.fc13.i686
--> Running transaction check
---> Package postgresql-devel.i686 0:8.4.4-1.fc13 set to be erased
---> Package postgresql-server.i686 0:8.4.4-1.fc13 set to be erased
--> Finished Dependency Resolution
Error: Trying to remove "postgresql", which is protected
You could try using --skip-broken to work around the problem
You could try running: rpm -Va --nofiles --nodigest
```

The error message is clear: the uninstall has failed. To be able to uninstall, it is necessary to remove the name of the package from **rdbsms.conf**.

Yum Command Usage

After configuring the YUM system, you can use the **yum** command line (CLI). It will be demonstrated with examples using common useful parameters necessary in the daily operations of a Fedora system administrator.

5.1. Yum Commands and Syntax

If you check the man pages, **man yum**, you will find the following definition for the use of **yum**:

```
yum [options] [command] [package ...]
```

From the syntax we know that there are **yum** options and commands, followed by a list of package names. Different **yum** commands and options will be mixed throughout the development of this manual, but it is very important to emphasize their difference and that not all options apply to all commands. The main reference is the **yum** manual page yum (**man yum**). One quick option is simply typing the **yum** command and see the usage help it displays, for example, below is the output of **yum** according to the extensions installed on a given Fedora 13 system, listing the commands and options:

```
$ yum
Loaded plugins: aliases, filter-data, list-data, local, merge-conf, presto,
               : refresh-packagekit, tmprepo
You need to give some command
Usage: yum [options] COMMAND

List of Commands:

alias                Adds or lists aliases
check                Check for problems in the rpmdb
check-update         Check for available package updates
clean                Remove cached data
deplist              List a package's dependencies
distribution-synchronization Synchronize installed packages to the latest available versions
downgrade            downgrade a package
erase                Remove a package or packages from your system
groupinfo            Display details about a package group
groupinstall         Install the packages in a group on your system
grouplist             List available package groups
groupremove          Remove the packages in a group from your system
help                 Display a helpful usage message
history              Display, or use, the transaction history
info                 Display details about a package or group of packages
info-arches          Display aggregate data on the arch attribute of a group of packages
And list all the packages under each
info-archive-sizes   Display aggregate data on the archivesize attribute of a group of packages
And list all the packages under each
info-baseurls        Display aggregate data on the url attribute of a group of packages
And list all the packages under each
info-buildhosts      Display aggregate data on the buildhost attribute of a group of packages
And list all the packages under each
info-committers      Display aggregate data on the committer attribute of a group of packages
And list all the packages under each
info-groups          Display aggregate group data, for matching members
And list all the packages under each
info-installed-sizes Display aggregate data on the installedsize attribute of a group of
packages
And list all the packages under each
info-licenses        Display aggregate data on the license attribute of a group of packages
And list all the packages under each
info-package-sizes   Display aggregate data on the packagesize attribute of a group of packages
```

```
And list all the packages under each
info-packagers Display aggregate data on the packager attribute of a group of packages
And list all the packages under each
info-rpm-groups Display aggregate data on the group attribute of a group of packages
And list all the packages under each
info-vendors Display aggregate data on the vendor attribute of a group of packages
And list all the packages under each
install Install a package or packages on your system
list List a package or groups of packages
list-arches Display aggregate data on the arch attribute of a group of packages
list-archive-sizes Display aggregate data on the archivesize attribute of a group of packages
list-baseurls Display aggregate data on the url attribute of a group of packages
list-buildhosts Display aggregate data on the buildhost attribute of a group of packages
list-committers Display aggregate data on the committer attribute of a group of packages
list-groups Display aggregate group data, for matching members
list-installed-sizes Display aggregate data on the installedsize attribute of a group of
packages
list-licenses Display aggregate data on the license attribute of a group of packages
list-package-sizes Display aggregate data on the packagesize attribute of a group of packages
list-packagers Display aggregate data on the packager attribute of a group of packages
list-rpm-groups Display aggregate data on the group attribute of a group of packages
list-vendors Display aggregate data on the vendor attribute of a group of packages
localinstall Install a local RPM
makecache Generate the metadata cache
provides Find what package provides the given value
reinstall reinstall a package
repolist Display the configured software repositories
resolvedep Determine which package provides the given dependency
search Search package details for the given string
shell Run an interactive yum shell
update Update a package or packages on your system
upgrade Update packages taking obsoletes into account
version Display a version for the machine and/or available repos.
```

Options:

```
-h, --help show this help message and exit
-t, --tolerant be tolerant of errors
-C, --cacheonly run entirely from system cache, don't update cache
-c [config file], --config=[config file]
config file location
-R [minutes], --randomwait=[minutes]
maximum command wait time
-d [debug level], --debuglevel=[debug level]
debugging output level
--showduplicates show duplicates, in repos, in list/search commands
-e [error level], --errorlevel=[error level]
error output level
--rpmverbosity=[debug level name]
debugging output level for rpm
-q, --quiet quiet operation
-v, --verbose verbose operation
-y, --assumeyes answer yes for all questions
--version show Yum version and exit
--installroot=[path] set install root
--enablerepo=[repo] enable one or more repositories (wildcards allowed)
--disablerepo=[repo] disable one or more repositories (wildcards allowed)
-x [package], --exclude=[package]
exclude package(s) by name or glob
--disableexcludes=[repo]
disable exclude from main, for a repo or for
everything
--obsoletes enable obsoletes processing during updates
--noplugins disable Yum plugins
--nogpgcheck disable gpg signature checking
--disableplugin=[plugin]
disable plugins by name
```

```

--enableplugin=[plugin]
    enable plugins by name
--skip-broken
    skip packages with depsolving problems
--color=COLOR
    control whether color is used
--releasever=RELEASEVER
    set value of $releasever in yum config and repo files
--setopt=SETOPTS
    set arbitrary config and repo options
--filter-vendors=FILTER_VENDORS
    Filter to packages with a matching vendor
--filter-rpm-groups=FILTER_RPM_GROUPS
    Filter to packages with a matching group
--filter-packagers=FILTER_PACKAGERS
    Filter to packages with a matching packager
--filter-licenses=FILTER_LICENSES
    Filter to packages with a matching license
--filter-committers=FILTER_COMMITTERS
    Filter to packages with a matching committer
--filter-arches=FILTER_ARCHES
    Filter to packages with a matching arch
--filter-buildhosts=FILTER_BUILDHOSTS
    Filter to packages with a matching buildhost
--filter-urls=FILTER_URLS
    Filter to packages with a matching url
--filter-package-sizes=FILTER_PACKAGE_SIZES
    Filter to packages with a packagesize in the given
    range
--filter-archive-sizes=FILTER_ARCHIVE_SIZES
    Filter to packages with a archivesize in the given
    range
--filter-installed-sizes=FILTER_INSTALLED_SIZES
    Filter to packages with a installedsize in the given
    range
--filter-groups=FILTER_GROUPS
    Filter to packages within a matching yum group
--merge-conf
    Merge configuration changes after installation
--disablepresto
    disable Presto plugin and don't download any deltarpm's

Plugin Options:
--tmprepo=[url]
    enable one or more repositories from URLs
--tmprepo-keep-created
    keep created direcotry based tmp. repos.

```

5.2. Packages names

When you need to refer to a package name to install, update, remove, list, get information, etc., you can specify the package name with one of the following formats:

- name
- name.arch
- name-ver
- name-ver-lib
- name-ver-lib.arch
- name-epoch:ver-lib.arch
- epoch:name-ver-lib.arch

For example:

```
# yum remove kernel-2.4.1-10.i686
```

It will remove the kernel of that specific version and architecture.

```
# yum list available 'foo*'
```

It will list all available packages that match the 'foo *' pattern. Single quotes prevent shell expansion.

```
Architecture Compatibility
i386 Suitable for any current Intel-compatible computer
noarch Compatible with all computer architectures
ppc Suitable for PowerPC systems, such as Apple Power Macintosh
x86_64 Suitable for 64-bit Intel-compatible processors, such as Opterons
```

Use the short name in **yum** commands. This causes **yum** to automatically select the latest package in the repositories that match your computer architecture.

Specify the name with any of the other formats to override the default **yum** behavior and force it to use the package that match your specification either with version or with architecture. Make this decision only when you know that the default selection has a bug or some other failure that would turn your system unusable.

5.3. The Help Command

yum includes the **help** command to provide minimal online help and is used by adding the command for which you want help, some examples:

```
# yum help install
...
install PACKAGE...

Install one or several packages in your system
```

```
# yum help erase
...
erase PACKAGE...

Removes one or several packages from your system

alias: remove
```

5.4. Repository Listing

Whenever enabled repositories are referred to in this manual, it is important to identify them from the command line without having to manually review the .repo files. The **yum** command has an adequate subcommand **repolist** for that task with its respective options.

```
# yum repolist enabled
...
repository id      repository name      state
chromium           Chromium Test Packages      13
fedora             Fedora 12 - i386          15366
google-chrome      google-chrome            5
playonlinux        PlayOnLinux Official repo   7
poptop-stable      PoPToP stable repository for Fedora Core 12 (i  4
pptp-stable        PPTP Client stable repository for Fedora Core  6
rpmfusion-free     RPM Fusion for Fedora 12 - Free      383
```


rpmfusion-free-updates	RPM Fusion for Fedora 12 - Free - Updates	1287
rpmfusion-nonfree	RPM Fusion for Fedora 12 - Nonfree	124
rpmfusion-nonfree-updates	RPM Fusion for Fedora 12 - Nonfree - Updates	502
updates	Fedora 12 - i386 - Updates	6975
repolist: 24672		

Also you may list the disabled repositories or simply all of them:

```
# yum repolist disabled
...
repository id                repository name
adobe-linux-i386             Adobe Systems Incorporated
atrpms                       Fedora 12 - i386 - ATrpms
fedora-debuginfo             Fedora 12 - i386 - Debug
fedora-dvd                   Fedora 10 DVD - i386
fedora-source                 Fedora 12 - Source
...
rpmfusion-nonfree-updates-testing-debuginfo RPM Fusion for Fedora 12 - Nonfree
rpmfusion-nonfree-updates-testing-source    RPM Fusion for Fedora 12 - Nonfree
thatfleminggnt               That Fleming Gent's Repository for
thatfleminggnt-sources        That Fleming Gent's Repository for
updates-debuginfo             Fedora 12 - i386 - Updates - Debug
updates-source                Fedora 12 - Updates Source
updates-testing                Fedora 12 - i386 - Test Updates
updates-testing-debuginfo      Fedora 12 - i386 - Test Updates Deb
updates-testing-source         Fedora 12 - Test Updates Source
repolist: 0
```

```
# yum repolist all
...
repository id                repository name state
adobe-linux-i386             Adobe Systems Incor disabled
atrpms                       Fedora 12 - i386 - disabled
chromium                     Chromium Test Packa enabled: 13
fedora                       Fedora 12 - i386 enabled: 15,366
fedora-debuginfo             Fedora 12 - i386 - disabled
fedora-dvd                   Fedora 10 DVD - i38 disabled
fedora-source                 Fedora 12 - Source disabled
...
updates                       Fedora 12 - i386 - enabled: 6,975
updates-debuginfo            Fedora 12 - i386 - disabled
updates-source                Fedora 12 - Updates disabled
updates-testing                Fedora 12 - i386 - disabled
updates-testing-debuginfo      Fedora 12 - i386 - disabled
updates-testing-source         Fedora 12 - Test Up disabled
repolist: 24,672
```

5.5. Installing New Software

You should use the **yum install** command to install the latest version of a package or package group and at the same time ensuring that all dependencies are satisfied in the process.

You must specify the desired package names, if it does not succeed with the specified names, it will assume that these names are "shell glob " and any matching obtained from the repositories will be installed.

If it fails to identify the package, then a search will be performed.

5.5.1. Install from the enabled repositories

```
# su -c "yum install xfig"
```

```
...
--> Running transaction check
---> Package xfig.x86_64 0:3.2.5-25.b.fc14 set to be installed
--> Processing Dependency: xfig-common = 3.2.5-25.b.fc14 for package:
xfig-3.2.5-25.b.fc14.x86_64
--> Processing Dependency: libXaw3d.so.7()(64bit) for package: xfig-3.2.5-25.b.fc14.x86_64
--> Running transaction check
---> Package Xaw3d.x86_64 0:1.5E-19.fc14 set to be installed
---> Package xfig-common.x86_64 0:3.2.5-25.b.fc14 set to be installed
--> Processing Dependency: transfig >= 1:3.2.5 for package: xfig-
common-3.2.5-25.b.fc14.x86_64
--> Running transaction check
---> Package transfig.x86_64 1:3.2.5d-1.fc14 set to be installed
--> Finished Dependency Resolution
```

Dependencies Resolved

```
=====
Package                Arch          Version              Repository           Size
=====
Installing:
xfig                   x86_64        3.2.5-25.b.fc14      updates              506 k
Installing for dependencies:
Xaw3d                  x86_64        1.5E-19.fc14         updates              153 k
transfig               x86_64        1:3.2.5d-1.fc14      updates              276 k
xfig-common            x86_64        3.2.5-25.b.fc14      updates              4.6 M
```

Transaction Summary

```
=====
Install                4 Package(s)
```

Total download size: 5.5 M

Installed size: 15 M

Is this ok [y/N]: Downloading Packages:

Setting up and reading Presto delta metadata

Processing delta metadata

Package(s) data still to download: 5.5 M

```
-----
Total                                107 kB/s | 5.5 MB    00:53
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing      : 1:transfig-3.2.5d-1.fc14.x86_64                1/4
  Installing      : Xaw3d-1.5E-19.fc14.x86_64                    2/4
  Installing      : xfig-common-3.2.5-25.b.fc14.x86_64            3/4
  Installing      : xfig-3.2.5-25.b.fc14.x86_64                  4/4
```

Installed:

xfig.x86_64 0:3.2.5-25.b.fc14

Dependency Installed:

Xaw3d.x86_64 0:1.5E-19.fc14 transfig.x86_64 1:3.2.5d-1.fc14

xfig-common.x86_64 0:3.2.5-25.b.fc14

Complete!

In the previous example we want to install **xfig** but a number of uninstalled dependencies are needed.

yum has detected this and it has included the dependencies needed to succeed in the task.

5.5.2. Finger error, look at the suggestions

You may want to install a software package but you do not know its package name in Fedora or finger error is made when specifying the desired package name, in these cases **yum** includes a search and its results as suggestions:

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

Note the suggestion that perhaps you wanted to mean "Zim" instead of "zim". Also note that the package name identification is case sensitive.



Warning

With LANG=es_ES.UTF-8 it still does not show the suggestion.

```
# yum install zim
...
Configurando el proceso de instalación
No existe disponible ningún paquete zim.
Nada para hacer
```

5.5.3. Installing an RPM

In case you are provided of an RPM file, you can use the same **yum install** command, which is equivalent to the obsolete **yum localinstall** command, as follows:

```
# su -c "yum localinstall psad-2.1.4-1.i386.rpm"
```

or

```
# su -c "yum install psad-2.1.4-1.i386.rpm"
```

In this way, if there are dependencies, **yum** will detect and install them. An additional option is often necessary, **--nogpgcheck**, for cases when you create your own RPM and do not sign the packages or you have not imported the necessary gpg key to the key ring.

```
# su -c "yum --nogpgcheck install psad-2.1.4-1.i386.rpm"
```



Request Signed Packages

If the provider does not sign the package is not really recommended to use them unless you have absolute confidence of the source of the package, require to software vendors to always sign their packages for security reasons and import the necessary signature.

```
# su -c "rpm --import /ruta/a_la/llave/GPG"
```

The **yum localinstall** command is deprecated, use the form **yum install**.

5.5.4. Reinstalling Software

The **yum reinstall** command will reinstall exactly the same currently installed version of the package without considering the updated packages. This does not work with "installonly" packages such as the kernels. **yum reinstall** works on groups, files, provided files and file lists too, just like **yum install** command. This command is useful in cases in which the package installation has been corrupted, for example, when you have lost the manual or the binaries have been removed accidentally.

```
# yum check-update yum-plugin-changelog
...
yum-plugin-changelog                1.1.28-1.fc13                updates

# yum list installed yum-plugin-changelog
...
yum-plugin-changelog.noarch         1.1.26-1.fc13                @fedora

# yum reinstall yum-plugin-changelog
...                               | 3.2 kB    00:00 ...

Dependencies Resolved

=====
Package                Arch      Version              Repository            Size
=====
Reinstalling:
yum-plugin-changelog    noarch    1.1.26-1.fc13        fedora                 17 k

Transaction Summary
=====
Reinstall      1 Package(s)

Total download size: 17 k
Installed size: 22 k
Is this ok [y/N]:
...

# yum list installed yum-plugin-changelog
...
Paquetes instalados
yum-plugin-changelog.noarch         1.1.26-1.fc13                @fedora
```

Note that in this procedure still exists a new version and exactly the same previous version is reinstalled, this is actually the **yum reinstall** functionality.

5.6. Updating Software

In general **yum update** command is required to try to update a package or group of packages already installed. The **yum update** verifies that the packages to be updated are installed, this is a difference in behavior shown by **rpm -U paquete.rpm** command.

```
# yum update zzuf
Loaded plugins:aliases, downloadonly, fastestmirror, list-data,
...
Setting up Update Process
Package zzuf available but not installed.
No package selected for update
```

5.6.1. Updating a Package

This **yum** option is natural and intuitive as previously noted, see the following simple example of use specifying a package to update.

```
# su -c "yum update xfig"
```

5.6.2. Updating with an RPM

This case is similar to installing from an rpm file, in this case an rpm file is provided to update a previously installed version.

```
# su -c "yum update psad-2.1.4-1.i386.rpm"
```

or

```
# su -c "yum localupdate psad-2.1.4-1.i386.rpm"
```

The **yum localupdate** form is obsolete, use **yum update** from now on.

5.6.3. Downgrade to a Previous Version

yum downgrade is available to allow to go back to the previous version of the specified software available in the repositories.

```
# rpm -q vorbis-tools
vorbis-tools-1.2.0-8.fc12.i686
# yum downgrade vorbis-tools
...
Resolving Dependencies
--> Running transaction check
---> Package vorbis-tools.i686 1:1.2.0-6.fc12 set to be updated
---> Package vorbis-tools.i686 1:1.2.0-8.fc12 set to be erased
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                Arch          Version      Repository    Size
=====
Downgrading:
vorbis-tools           i686          1:1.2.0-6.fc12      fedora        157 k

Transaction Summary
=====
Erase                0 Package(s)
Reinstall            0 Package(s)
Downgrade            1 Package(s)

Total download size: 157 k
Is this ok [y/N]:y
Downloading Packages:
Setting up and reading Presto delta metadata
Processing delta metadata
Package(s) data still to download: 157 k
vorbis-tools-1.2.0-6.fc12.i686.rpm                157 kB      00:01
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
```

```
Installing      : 1:vorbis-tools-1.2.0-6.fc12.i686      1/2
Cleaning       : 1:vorbis-tools-1.2.0-8.fc12.i686      2/2

Erased:
  vorbis-tools.i686 1:1.2.0-8.fc12
Installed:
  vorbis-tools.i686 1:1.2.0-6.fc12
Complete!
# rpm -q vorbis-tools
vorbis-tools-1.2.0-6.fc12.i686
```



Use in conjunction with repo updates-testing

The [updates] repository only keeps the last version of the package, so to use **yum downgrade** reliably you should enable [updates-testing] repository and issue a **yum update ...** and then, if it does not work, you can rollback to the previous version in [updates] or in [fedora] repo with **yum downgrade**

Another alternative is to keep a local repository with these versions with **yum-plugin-local** extension.

5.6.4. General Update

If you want to update the system and all installed applications, use the following command:

```
# su -c "yum update"
```

5.6.5. Checking and Update Listing

This **yum** command lets you know if the system has pending updates to be applied.

```
# su -c "yum check-update"
...

finch.i686                2.7.2-1.fc12          updates
git.i686                  1.7.2-1.fc12          updates
iftop.i686                0.17-11.fc12         updates
libpurple.i686            2.7.2-1.fc12          updates
perl-Git.noarch           1.7.2-1.fc12          updates
pidgin.i686               2.7.2-1.fc12          updates
schroedinger.i686         1.0.8-4.fc12          updates
```

You can ask for updates for a specific package.

```
# su -c "yum check-update yum-plugin-changelog"
...
yum-plugin-changelog      1.1.28-1.fc13         updates
# su -c "yum list installed yum-plugin-changelog"
...
yum-plugin-changelog.noarch 1.1.26-1.fc13         @fedora
```

5.6.6. Exclude when Updating

Sometimes it is necessary to exclude a package during the upgrade process, you may want to keep the current version, later we will show another technique suitable for this particular situation. For now you may use the option **--exclude** in the following way:

```
# su -c "yum update --exclude=ekiga"
```

This may be necessary because at some point the mirrors used are not updated fast enough and you do not want to or can not wait, or just do not want to or can not use the newer version.

To exclude packages so that they can not be updated permanently by **yum**, you must modify **yum.conf** and add the directive *exclude*. You can use a package names list separated by commas.

```
[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
exclude=pidgin, amsn, emesene, paquete
```

You can also exclude packages from specific repositories, adding the exclude in the corresponding **.repo** file located in **/etc/yum.repos.d/nombre_del_repo.repo**

5.6.7. Avoiding packages with problems when updating

In certain situations you may get dependency problems that may break a software update process, if you want to upgrade avoiding these packages and their dependencies, use the **--skip-broken** option. This will not solve the dependency problems, but at least have the system as current as possible until a solution is given for the packages with problems.

```
# su -c "yum update --skip-broken"
...
```

5.7. Removing Software

Software uninstalling is simple too, use **yum remove** command.

```
# su -c "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]:
```

You may also use the form:

```
# su -c "yum erase xifg"
...
```

5.8. Querying Package Information

You may view the details of a specific package with **yum info** command.

```
$ su -c "yum info squid"
Loaded plugins: aliases, filter-data, list-data, local, merge-conf, presto,
               : refresh-packagekit, tmprepo
http://download.fedoraproject.org/pub/fedora/linux/updates/14/x86_64/
repodata/698c6412d340528a58ae94da66dbfebde5b9650ee4ffaf92f80855002845afec-primary.sqlite.bz2:
[Errno -1] Metadata file does not match checksum
Trying other mirror.
Installed Packages
Name       : squid
Arch       : x86_64
Epoch     : 7
Version    : 3.1.9
Release    : 3.fc14
Size       : 5.7 M
Repo       : installed
From repo  : updates
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
              : especially 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.
```

If more than one packages matches the package name or glob, then probably **yum info** will show information about each one of them. In the example below a new version of yum is available in the repositories and it is not installed yet, **yum info** lists both packages and their respective information.

```
$ yum info yum
...
Installed Packages
Name       : yum
Arch       : noarch
Version    : 3.2.28
Release    : 3.fc13
Size       : 4.2 M
Repository : installed
From repo  : updates
Summary    : RMP installer/updater
URL        : http://yum.baseurl.org/
License    : GPLv2+
Description: Yum is a utility that can check for and automatically download and
              : install updated RPM packages. Dependencies are obtained and
              : downloaded automatically, prompting the user for permission as
              : necessary.

Available Packages
Name       : yum
Arch       : noarch
```



```

Version      : 3.2.28
Release      : 4.fc13
Size         : 913 k
Repository   : updates
Summary      : RPM installer/updater
URL          : http://yum.baseurl.org/
License      : GPLv2+
Description  : Yum is a utility that can check for and automatically download and
              : install updated RPM packages. Dependencies are obtained and
              : downloaded automatically, prompting the user for permission as
              : necessary.

```

Root only

The **yum info** listing will not show the other results if it is not run with root privileges.

5.9. Queries and Package Listings

This section will review package searching and listing methods with **yum**.

5.9.1. Search by Name, Description and Summary

yum search command provides a package listing with results of the search by pattern matching in the description, summary and package name. Let us see an example:

```

$ su -c "yum search imap"
...
===== Matched: imap =====
cyrus-imapd.x86_64 : A high-performance mail server with IMAP, POP3, NNTP and
                  : SIEVE support
cyrus-imapd-devel.i686 : Cyrus IMAP server development files
cyrus-imapd-devel.x86_64 : Cyrus IMAP server development files
cyrus-imapd-utils.x86_64 : Cyrus IMAP server administration utilities
imapfilter.x86_64 : A flexible client side mail filtering utility for IMAP
                  : servers
imapsync.noarch : Tool to migrate email between IMAP servers
...
...
...
squirrelmail.noarch : webmail client written in php
stunnel.x86_64 : An SSL-encrypting socket wrapper
tcpjunk.x86_64 : TCP protocols testing tool
wammu.noarch : Mobile Phone Manager - Gammu GUI
xfce4-mailwatch-plugin.x86_64 : Mail Watcher plugin for the Xfce panel
zarafa.x86_64 : Open Source Edition of the Zarafa Collaboration Platform

```

5.9.2. Search by file or functionality

yum provides command or its alias **yum whatprovides** is another way to search the package database. In this case, it searches for the package that provide some functionality or package file. Use a specific name or a file-glob wildcards syntax to list the available or installed packages that provides the functionality or desired file.

```

$ su -c "yum provides /etc/httpd/conf/httpd.conf"
...
httpd-2.2.16-1.fc14.x86_64 : Apache HTTP Server
Repo          : fedora
Matched from:
Filename      : /etc/httpd/conf/httpd.conf

```

```
httpd-2.2.17-1.fc14.x86_64 : Apache HTTP Server
Repo                        : updates
Matched from:
Filename                   : /etc/httpd/conf/httpd.conf

httpd-2.2.17-1.fc14.x86_64 : Apache HTTP Server
Repo                        : installed
Matched from:
Other                       : Provides-match: /etc/httpd/conf/httpd.conf
```

5.9.3. Listing installed packages

yum list installed command shows a listing of the packages actually installed in the system.

```
# su -c "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
.... [many others]
```

5.9.4. Listing of available packages

It is possible to list the available packages for installation from the configured repositories:

```
# su -c "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
....
[many others]
```

5.9.5. Listing available updates

It is possible to list the available updates for installed packages from the enabled repositories:

```
# su -c "yum list updates"
...
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
... [others]
```

5.9.6. Listing recent packages

It is possible to list the packages that have been recently added to the configured repositories by using the **recent** option of the **list** command:

```
# su -c "yum list recent" | head -n 15
...
Recently Added Packages
ditaa.noarch                0.9-4.r74.fc12          updates
extundelete.i686           0.2.0-1.fc12            updates
frama-c.i686                1.4-5.fc12              updates
frama-c-devel.i686         1.4-5.fc12              updates
google-chrome-beta.i386    5.0.375.125-53311       google-chrome
google-chrome-stable.i386  5.0.375.125-53311       google-chrome
google-chrome-unstable.i386 6.0.472.14-54133        google-chrome
google-desktop-linux.i386   1.2.0-0088              google-chrome
gphotoframe.noarch         1.0-2.fc12              updates
gphotoframe-gss.noarch     1.0-2.fc12              updates
```

5.9.7. Listing foreign packages

This listing can be very important since it lists all installed packages not available in the configured and enabled repositories. Probably an rpm was downloaded from a third party web server and installed, a foreign package for the enabled repositories. It may also happen that some old packages are still alive without your knowledge in your system because some kind of dependency problems. It is also possible that you've enabled updates-testing repository, or another one, and temporarily installed some package from that repository and then disabled it.

```
# su -c "yum list extras"
Extra Packages
AdobeReader_sve.i486        8.1.7-1                 @adobe-linux-i386/12
adobe-release-i386.noarch    1.0-1                   installed
bce-devel.x86_64            1.8-1.fc13              @/bce-devel-1.8-1.fc13.x86_64/13
cherokee.x86_64             1.0.6-1.fc13            @updates/13
dbus-qt.x86_64              0.70-7.fc12             @fedora/12
exif.x86_64                 0.6.9-1                 installed
fox.x86_64                  1.6.38-1.fc13           @/fox-1.6.38-1.fc13.x86_64
gnome-applet-sensors.x86_64 2.2.7-3.fc13            @updates/13
ifd-egate.x86_64            0.05-22                 @anaconda-
InstallationRepo-200911081904.x86_64/12
kernel.x86_64               2.6.35.6-48.fc14        @updates
kernel.x86_64               2.6.35.9-64.fc14        @updates
kplayer.x86_64              1:0.7.0-3.20081211cvs.fc13
                             @rpmfusion-free-updates/13
libjpeg.i686                6b-46.fc12              @fedora/12
lohit-oriya-fonts.noarch    2.4.3-6.fc13            @updates/13
moc.x86_64                  2.4.3-3.fc12.mf         @thatfleminggnt/12
rubymgem-ruby-rpm.x86_64    1.3.0-1.fc13            @/rubymgem-ruby-rpm-1.3.0-1.fc13.x86_64/13
sendEmail.noarch            1.56-1.fc13             installed
skype.i586                  2.1.0.81-fc10           installed
thatfleminggnt-release.noarch 12-1.mf                 installed
vzctl-lib.x86_64            3.0.23-1                @openvz-utils/12
yum-plugin-appmarket.noarch 1.0-1                   @fedora-appmarket/13
zaz-debuginfo.x86_64        0.3.3-1.fc13            @fedora-debuginfo/13
```

5.10. Enable a repository on the go

It is common to have disabled repositories, for example, rpm sources or repositories with packages under testing, and given the circumstances it is necessary to enable them and run some **yum** command, for example, for searching for information on a package in rawhide, use **--enablerepo** on the go:

```
# su -c "yum info --enablerepo=rawhide bro"
...
rawhide                                | 4.3 kB    00:00
```

```
rawhide/primary_db | 13 MB 01:55
...
Available Packages
Name      : bro
Arch      : i686
Version   : 1.5.1
Release   : 2.fc15
Size      : 1.9 M
Repo      : rawhide
Summary   : Open-source, Unix-based Network Intrusion Detection System
URL       : http://bro-ids.org
License   : BSD
Description : Bro is an open-source, Unix-based Network Intrusion Detection
            : System (NIDS) that passively monitors network traffic and looks
            : for suspicious activity. Bro detects intrusions by first parsing
            : network traffic to extract its application-level semantics and then
            : executing event-oriented analyzers that compare the activity with
            : patterns deemed troublesome. Its analysis includes detection of
            : specific attacks (including those defined by signatures, but also
            : those defined in terms of events) and unusual activities (e.g.,
            : certain hosts connecting to certain services, or patterns of
            : failed connection attempts).
```

5.11. Dependencies Listing

It is very easy to list the dependencies of a given package with **yum**.

```
# yum deplist rubygems
...
Finding dependencies:
package: rubygems.noarch 1.3.7-2.fc14
dependency: ruby-rdoc
  provider: ruby-rdoc.noarch 1.8.7.302-1.fc14
  provider: ruby-rdoc.noarch 1.8.7.302-2.fc14
dependency: ruby(abi) = 1.8
  provider: ruby-libs.x86_64 1.8.7.302-1.fc14
  provider: ruby-libs.i686 1.8.7.302-1.fc14
  provider: ruby-libs.x86_64 1.8.7.302-2.fc14
  provider: ruby-libs.i686 1.8.7.302-2.fc14
dependency: /usr/bin/ruby
  provider: ruby.x86_64 1.8.7.302-1.fc14
  provider: ruby.x86_64 1.8.7.302-2.fc14
```

5.12. Query the cache

If you need to query the package database and you do not have network connection with the enabled repositories, you may use the **-C** to only search on the local cache. The data in the cache must be up to date and the required action must not imply a download from the network. Simple example of a package search:

```
$ yum -C search squid
...
===== Matched: squid =====
lightsquid.noarch : Light, small, and fast log analyzer for squid proxy
lightsquid-apache.noarch : Web Controls for lightsquid
squid.x86_64 : The Squid proxy caching server
squidGuard.x86_64 : Filter, redirector and access controller plugin for squid
calamaris.noarch : Squid native log format (NLF) analyzer and report generator
awstats.noarch : Advanced Web Statistics
ccze.x86_64 : A robust log colorizer
dansguardian.x86_64 : Content filtering web proxy
mod_auth_ntlm_winbind.x86_64 : NTLM authentication for the Apache web server
```

```

: using winbind daemon
tinyproxy.x86_64 : A small, efficient HTTP/SSL proxy daemon

```

5.13. Show duplicates

In normal operation, **yum** only considers the latest version of a given package. To view previous versions, you must enable the list of duplicates:

```

[main]
cachedir=/var/cache/yum
keepcache=0
debuglevel=2
logfile=/var/log/yum.log
gpgcheck=1
plugins=1
# Mostrar duplicados
showdupesfromrepos=1

```

An example of the results follows.

showdupesfromrepos=0

```

# su -c "yum list available vorbis-tools"
...
Available Packages
vorbis-tools.i686                1:1.2.0-8.fc12                updates

```

showdupesfromrepos=1

```

# su -c "yum list available vorbis-tools"
...
Available Packages
vorbis-tools.i686                1:1.2.0-6.fc12                fedora
vorbis-tools.i686                1:1.2.0-8.fc12                updates

```

It is also possible to only enable it on the go with **--showduplicates** option.

```

# su -c "yum list available vorbis-tools --showduplicates"
...
Available Packages
vorbis-tools.x86_64              1:1.2.0-7.fc13                fedora
vorbis-tools.x86_64              1:1.2.0-8.fc13                updates

```

5.14. Existing lock ...

If you get the next message when running a **yum** command:

```

# yum install xxx*
Existing lock /var/run/yum.pid: another copy is running as pid 4112.
Another app is currently holding the yum lock; waiting for it to exit...
  The other application is: yum
    Memory : 109 M RSS (412 MB VSZ)
    Started: Sat Dec 25 15:01:12 2010 - 00:31 ago
    State  : Sleeping, pid: 4112

```

It is because there is another **yum** instance running or dead. The origin of this situation is that the system does not support YUM concurrent operations. A circumstance that usually occur is when **yum-updatesd** is running. You may solve it with:

```
$ su -
password
# service yum-updatesd stop
<my yum commands>
# service yum-updatesd start
```

This only works if **yum-updatesd** is locking the system. If it is another **yum** application, then simply wait, and only as a last resort, if you think that the application is dead and unresponsive:

```
su -
password
kill -kill <pid>
```

Where <pid> is the process id of the application.

5.15. YUM Software Groups

Now we will review the **yum** commands and configurations related to software groups that are certainly very useful and refer to the **yum** system's ability to handle the listing, installing, upgrading or removing software groups in a single **yum** transaction.

The general syntax for **yum** commands available for groups is:

```
* groupinstall group1 [group2] [...]
* groupupdate group1 [group2] [...]
* grouplist [hidden] [groupwildcard] [...]
* groupremove group1 [group2] [...]
* groupinfo group1 [...]
```

Alternatively you can use the following base commands by specifying the group name prefixed by the "@" to indicate that this is a group name instead of a package name.

```
* install @group1 [@group2] [...]
* update @group1 [@group2] [...]
* remove @group1 [@group2] [...]
```

5.15.1. Listing available groups

For a listing of the available **YUM** software groups use the **grouplist** option.

```
# yum grouplist
...
Setting up Group Process
Installed Groups:
Administration Tools
Arabic Support
Armenian Support
Assamese Support
Authoring and Publishing
Base
...
...
Venda Support
Virtualization
Web Development
Web Server
Window Managers
```

```

Windows File Server
X Window System
Available Groups:
Afrikaans Support
Albanian Support
Amazigh Support
Asturian Support
...
Welsh Support
X Software Development
XFCE
XFCE Software Development
Xhosa Support
Zulu Support
Done

```

5.15.2. Packages info from a group

For obtaining the group info and its packages use the **groupinfo** option.

```

# yum groupinfo "GNOME Desktop Environment"
...
Setting up Group Process

Group: GNOME Desktop Environment
Description: GNOME is a powerful graphical user interface which includes a panel, desktop,
system icons, and a graphical file manager.
Mandatory Packages:
control-center
fedora-gnome-theme
gnome-panel
gnome-session
gnome-terminal
gvfs-fuse
metacity
nautilus
polkit-gnome
yelp
Default Packages:
NetworkManager-gnome
NetworkManager-openconnect
NetworkManager-openvpn
...
...
seahorse
shotwell
simple-scan
vinagre
vino
xdg-user-dirs-gtk
zenity
Optional Packages:
alacarte
avant-window-navigator
beagle-evolution
beagle-gnome
buoh
byzanz
...
...
tango-icon-theme
tango-icon-theme-extras
tomboy
tracker-search-tool
verbiste-gnome

```

```

wallpapoz
xiphos
xscreensaver-extras-gss
xscreensaver-gl-extras-gss
Conditional Packages:
  gtk3-engines
  ibus-gtk2
  ibus-gtk3
  libcanberra-gtk3
  pinentry-gtk
  scim-bridge-gtk

```

5.15.3. Installing and Removing Groups

Each yum software group contains a list of mandatory, optional and default packages. By applying the command to install the **yum** group, you will have to specify which of these packages will be installed. The default for this parameter is "default, mandatory", ie the components to be installed are those labeled as mandatory and as default. The configuration directive in **yum.conf** is *group_package_types*:

```

[main]
cachedir=/var/cache/yum
keepcache=0
debuglevel=2
logfile=/var/log/yum.log
...
# Setting to install every package from a group
group_package_types=default, mandatory, optional

```

Specifying which packages from the desired group can be installed, an example of group installation, in this case *Clustering*.

```

# yum groupinstall "Clustering" ...
Dependencies Resolved
=====
Package                               Arch          Version
Repository                            Size
=====
Installing:
cluster-cim                           x86_64        0.18.1-1.fc14
  fedora                               110 k
cluster-snmpp                         x86_64        0.18.1-1.fc14
  fedora                               126 k
heartbeat                             x86_64        3.0.0-0.7.0daab7da36a8.hg.fc14
  fedora                               172 k
ipvsadm                               x86_64        1.25-5.fc14
  fedora                               39 k
modcluster                            x86_64        0.18.1-1.fc14
  fedora                               195 k
pacemaker                             x86_64        1.1.4-4.fc14
  updates                             583 k
rgmanager                             x86_64        3.1.0-2.fc14
  updates                             155 k
ricci                                 x86_64        0.18.1-2.fc14
  updates                             639 k
Installing for dependencies:
OpenIPMI-libs                         x86_64        2.0.18-4.fc14
  fedora                               477 k
cifs-utils                            x86_64        4.7-1.fc14
  updates                             39 k
cluster-glue                          x86_64        1.0.6-1.fc14
  fedora                               238 k
...

```



```

resource-agents      x86_64      3.1.0-1.fc14
updates              394 k
sg3_utils            x86_64      1.29-1.fc14
fedora               475 k
tog-pegasus          x86_64      2:2.9.0-11.fc14
fedora               4.1 M

```

Transaction Summary

```
=====
```

```
Install      33 Package(s)
```

```
Total download size: 10 M
```

```
Installed size: 51 M
```

```
Is this ok [y/N]:
```

Obviously, finding out each and every one of the package names and install them one by one, even with its dependencies computed by **yum** it would have been a headache, demonstrating the usefulness of the **yum** groups in software management.

You may also update a previously installed group with, for example:

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

For removal, you must give the **groupremove**, for example, to uninstall all the GNOME desktop with only one command:

```
# yum groupremove "GNOME Desktop Environment"
```

Unlike the procedure made by **groupinstall**, the **groupremove** option will remove all packages regardless of their type, ie without taking into account value of *group_package_types* configuration option. It is important to note that a given package can be in more than one group at a time, so the sequence **groupinstall X Y** followed by **groupremove Y** is not necessarily the same to only **groupinstall X**.

5.16. YUM history and logs

yum supports the logging of its activity in a file given by the *logfile* directive in the *[main]* section of **yum.conf** file.

```
[main]
cachedir=/var/cache/yum
keepcache=0
debuglevel=2
logfile=/var/log/yum.log
...
```

A sample of this file is shown below.

```
# tail /var/log/yum.log
Jul 30 20:39:37 Installed: fedora-packager-0.5.1.0-1.fc12.noarch
Jul 31 06:28:07 Updated: libpurple-2.7.2-1.fc12.i686
Jul 31 06:28:08 Updated: finch-2.7.2-1.fc12.i686
Jul 31 06:28:09 Updated: schroedinger-1.0.8-4.fc12.i686
Jul 31 06:28:32 Updated: pidgin-2.7.2-1.fc12.i686
Jul 31 06:28:33 Updated: iftop-0.17-11.fc12.i686
Jul 31 06:28:33 Updated: perl-Git-1.7.2-1.fc12.noarch
Jul 31 06:28:39 Updated: git-1.7.2-1.fc12.i686
Jul 31 06:42:49 Erased: yum-plugin-allowdowngrade
Aug 02 11:26:10 Erased: iftop
```

5.16.1. History

The new **yum history** command is to fulfil the need of record and report what have been done in our system with its installed software. The user can view with this command what happened with recent transactions.

To enable this command, you must set the *history_record* boolean **yum.conf**. Enabling this functionality will require some extra space on the system and more processing time but it allows to get a lot of information of what has occurred over the time. This boolean is enabled by default, so there is no need to do nothing unless you want disable this feature.

```
$ yum history
...
ID      | Login user      | Date and time | Action(s) | Altered
-----|-----|-----|-----|-----
627 | Guillermo <gomix> | 2010-12-24 12:39 | Install | 4
626 | System <unset> | 2010-12-24 11:01 | Update | 18
625 | Guillermo <gomix> | 2010-12-24 10:58 | Downgrade | 18
624 | System <unset> | 2010-12-24 10:52 | Update | 19
623 | Guillermo <gomix> | 2010-12-23 18:09 | Erase | 1
622 | Guillermo <gomix> | 2010-12-23 16:29 | Downgrade | 2
621 | Guillermo <gomix> | 2010-12-23 16:20 | Erase | 2
620 | Guillermo <gomix> | 2010-12-23 16:19 | Erase | 1
619 | Guillermo <gomix> | 2010-12-23 16:18 | Erase | 1
618 | Guillermo <gomix> | 2010-12-23 16:16 | Erase | 1
617 | System <unset> | 2010-12-23 13:16 | E, I, U | 30 EE
616 | System <unset> | 2010-12-23 13:01 | Update | 10
615 | System <unset> | 2010-12-17 12:22 | O, U | 197 EE
614 | Guillermo <gomix> | 2010-12-16 17:41 | Install | 6
613 | System <unset> | 2010-12-14 15:12 | Update | 2
612 | System <unset> | 2010-12-10 09:45 | Update | 21
611 | Guillermo <gomix> | 2010-12-08 22:32 | Erase | 3
610 | Guillermo <gomix> | 2010-12-08 22:21 | Install | 3
609 | Guillermo <gomix> | 2010-12-08 22:19 | Install | 1
608 | Guillermo <gomix> | 2010-12-08 22:18 | Erase | 1
history list
```

For more information use the transaction ID with the **yum history info** command.

```
$ yum history info 607
...
Transaction ID : 607
Begin time      : Wed Dec 8 22:14:39 2010
Begin rpmdb     : 2542:b5d33dac8483e6aee941535236c8637f070969eb
End time        : 22:14:44 2010 (5 seconds)
End rpmdb       : 2543:2424f6ef25efd89347cd4a5cd5190885f8476774
User            : Guillermo <gomix>
Return-Code     : Success
Transaction performed with:
  Installed      rpm-4.8.1-5.fc14.x86_64
  Installed      yum-3.2.28-5.fc14.noarch
  Installed      yum-plugin-aliases-1.1.28-1.fc14.noarch
Packages Altered:
  Install yum-plugin-local-1.1.28-1.fc14.noarch
history info
```

5.16.2. Undo? Redo!

One of the most valuable and definitely one of the most useful aspects of the **history** command is the ability to undo or redo a transaction. Below is an example of its use:

```
# yum history undo 1
```

```

...
Undoing transaction 1, from Fri Nov 27 22:39:59 2009
  Install      0xFFFF-0.3.9-4.fc12.i686
Resolving Dependencies
--> Running transaction check
---> Package 0xFFFF.i686 0:0.3.9-4.fc12 set to be erased
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                Arch          Version      Repository    Size
=====
Removing:
  0xFFFF              i686         0.3.9-4.fc12      installed     100 k

Transaction Summary
=====
Remove                1 Package(s)
Reinstalling          0 Package(s)
Downgrade             0 Package(s)

Is this ok [y/N]:y
Downloading Packages:
Running rpm_check_debug
Running transaction check
Running transaction
  Erasing      : 0xFFFF-0.3.9-4.fc12.i686

                                                    1/1

Erased:
  0xFFFF.i686 0:0.3.9-4.fc12

Complete!

```

The contents of the history must have changed:

```

# yum history
...
ID      | Login user          | Date and time   | Action(s)      | Altered
-----
      2 | root <root>         | 2009-11-27 23:07 | Erase          |      1
      1 | root <root>         | 2009-11-27 22:39 | Install        |      1

```

And now the transaction can be done again.

```

# yum history redo 1
...
Repeating transaction 1, from Fri Nov 27 22:39:59 2009
  Install      0xFFFF-0.3.9-4.fc12.i686
...

```

5.16.3. Corrupt History

It is possible to corrupt the history database, and in such cases you may see something like:

```

# yum history info 10
...
Transaction ID : 10
Initial date   : Fri Nov 27 12:34:18 2009
Initla Rpmdb   : 2005:ea899bf5e2243b770e23188d4dccbb78b4fe96b4
Final date     :                12:34:54 2009 (36 segundos)
Final Rpmdb    : 2011:0edcefb8dd72a2b498a1dad6dc89478bb8a1a85

```

```
User      : Traceback (most recent call last):
File "/usr/bin/yum", line 29, in <module>
  yummain.user_main(sys.argv[1:], exit_code=True)
...
```

At the time of writing there is not a known method for restoring it, so it is advisable to keep a backup of the file containing the history. If you do not have a backup you must create a new file for the history.

```
# yum history new
...
history new
```

When creating a new history, the older one is not really lost, and for hackers, these files are SQLite3 databases. Anyway, this database consists of only one file, and making a backup in a safe place is an easy task.

```
# cd /var/lib/yum/history
# ls -l
total 416
-rw-r--r-- 1 root root 407552 nov 27 12:34 history-2009-11-25.sqlite
-rw-r--r-- 1 root root 11264 nov 27 22:40 history-2009-11-27.sqlite
```

5.16.4. Responsible Identification with sudo

One of the disadvantages of `/var/log/yum.log` log file is that it does not identify the person responsible for the activity yum. Combining the use of **sudo** with **yum history** command it provides a more detailed information of the **yum** transaction. Below is sample output.

User gomix authorized via **sudo** to use the **yum** command.

```
[gomix@fricky ~]$ id
uid=502(gomix) gid=502(gomix) grupos=497(pulse-rt),502(gomix)
[gomix@fricky ~]$ sudo yum -y install iftop
...
```

Output and verification with **yum history**:

```
$ yum history | head
...
ID      | Login user          | Date and time    | Action(s)         | Altered
-----|-----|-----|-----|-----
 627 | Guillermo <gomix>   | 2010-12-24 12:39 | Install           | 4
 626 | System <unset>      | 2010-12-24 11:01 | Update            | 18
 625 | Guillermo <gomix>   | 2010-12-24 10:58 | Downgrade         | 18
 624 | System <unset>      | 2010-12-24 10:52 | Update            | 19
 623 | Guillermo <gomix>   | 2010-12-23 18:09 | Erase             | 1
 622 | Guillermo <gomix>   | 2010-12-23 16:29 | Downgrade         | 2
```

Details of the last transaction.

```
$ yum history info 627
...
Transaction ID : 627
Begin time     : Fri Dec 24 12:39:33 2010
Begin rpmdb    : 2543:79c064669327b059106d868335f372e1c79bade3
End time       : 12:40:04 2010 (31 seconds)
End rpmdb      : 2547:5c1b298973c681d16f6a1890092338698844a497
User           : Guillermo <gomix>
```

```

Return-Code      : Success
Transaction performed with:
  Installed      rpm-4.8.1-5.fc14.x86_64
  Installed      yum-3.2.28-5.fc14.noarch
  Installed      yum-plugin-aliases-1.1.28-1.fc14.noarch
  Installed      yum-presto-0.6.2-2.fc14.noarch
Packages Altered:
  Dep-Install    Xaw3d-1.5E-19.fc14.x86_64
  Dep-Install    transfig-1:3.2.5d-1.fc14.x86_64
  Install        xfig-3.2.5-25.b.fc14.x86_64
  Dep-Install    xfig-common-3.2.5-25.b.fc14.x86_64
history info

```

5.16.5. Checking the rpm database

yum check checks the local rpmdb database and displays information about any problem it may find.

```

# yum check
...
ImageMagick-c++-6.5.4.7-3.fc12.i686 has missing requires of ImageMagick = ('0', '6.5.4.7',
'3.fc12')
ImageMagick-perl-6.5.4.7-3.fc12.i686 has missing requires of ImageMagick = ('0', '6.5.4.7',
'3.fc12')
abrt-1.1.1-2.fc12.i686 has missing requires of librpm.so.0
abrt-1.1.1-2.fc12.i686 has missing requires of librpmio.so.0
baekmuk-ttf-batang-fonts-2.2-22.fc12.noarch has missing requires of baekmuk-ttf-fonts-common
= ('0', '2.2', '22.fc12')
baekmuk-ttf-dotum-fonts-2.2-22.fc12.noarch has missing requires of baekmuk-ttf-fonts-common =
('0', '2.2', '22.fc12')
baekmuk-ttf-fonts-ghostscript-2.2-22.fc12.noarch has missing requires of baekmuk-ttf-gulim-
fonts = ('0', '2.2', '22.fc12')

```


Yum extensions

yum application supports what are called *extensions* or *plugins*. These extensions add extra functionality to **yum** which are not part of core functionality. Most of them are available in Fedora, below is a short list and review of each one of them emphasizing their use and applicability.

```
yum-plugin-aliases
yum-plugin-changelog
yum-plugin-fastestmirror
yum-plugin-filter-data
yum-plugin-list-data
yum-plugin-verify
yum-presto
yum-plugin-auto-update-debug-info
yum-plugin-basearchonly
yum-plugin-downloadonly
yum-plugin-fs-snapshot
yum-plugin-keys
yum-plugin-local
yum-plugin-merge-conf
yum-plugin-post-transaction-actions
yum-plugin-priorities
yum-plugin-protect-packages
yum-plugin-protectbase
yum-plugin-refresh-updatesd
yum-plugin-remove-with-leaves
yum-plugin-rpm-warm-cache
yum-plugin-security
yum-plugin-show-leaves
yum-plugin-tmprepo
yum-plugin-tsflags
yum-plugin-upgrade-helper
yum-plugin-versionlock
yum-rhn-plugin
PackageKit-yum-plugin
anaconda-yum-plugins
```

In this manual we discuss some extensions, for more information about other of them see <http://yum.baseurl.org/wiki/YumUtils>.

6.1. Extension installation and general configuration

The central site for extension configuration in Fedora is the directory `/etc/yum/pluginconf.d`.

In general, and in order to not repeat it for each extension, the command to install an extension is **yum install extension-name**, for example:

```
# su -c "yum install yum-plugin-downloadonly"
...
Dependencies Resolved
=====
Package                Arch    Version      Repository Size
=====
Installing:
yum-plugin-downloadonly noarch  1.1.24-2.fc12 updates  8.8 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
```

The name of each subsequent section will indicate the extension/package name to install according to its name in the current version of Fedora at the time this manual was written.

6.2. yum-plugin-aliases

This extension adds the **alias** command which allows to configure and use alias for **yum** commands.

yum alias [ALIAS] [expansion]

Add an alias or list them.

6.2.1. Configuration

/etc/yum/pluginconf.d/aliases.conf

```
[main]
enabled=1
conffile=/etc/yum/aliases.conf
recursive=true
register=false
```

enabled = 0|1

Enable extension at a general level.

conffile = /path

Configuration file for aliases.

recursive = true|false

Recursion use in alias.

register = true|false

Register alias for commands in yum.

The file **/etc/yum/aliases.conf** already incorporates a number of common aliases and may be freely edited with any text editor with the format *alias command* where if recursion is enabled, *command* may in turn contain other aliases.

Shows the alias configuration file as included in Fedora, **/etc/yum/aliases.conf**.

```
# Here are some aliases that many people find useful
# Note that you need to have recursion enabled

UPT --enablerepo=updates-testing
DUPS --showduplicates

SRC --enablerepo=fedora-source --enablerepo=updates-source
DEVSRG --enablerepo=development-source
UPTSRC --enablerepo=updates-testing-source

ALLSRC SRC DEVSRG UPTSRC

up upgrade
inst install
in install
rm remove
down downgrade
```



```
rein reinstall

chk check-update

ls list
lsi ls installed
lsa ls available
lsu ls updates
lupi list-updateinfo

h history
hi history info
hl history list
hs history summary

v version
```

6.2.2. Usage examples

Below **ls** is an alias for **list**, **lsi** is an alias for **ls installed** and **lsu** is an alias for **list updates**.

Lists all packages, whether installed or available.

```
# yum ls
...
zziplib.x86_64          0.13.49-8.fc12      fedora
zziplib-devel.i686     0.13.49-8.fc12      fedora
zziplib-devel.x86_64   0.13.49-8.fc12      fedora
zziplib-utils.x86_64   0.13.49-8.fc12      fedora
zzuf.x86_64            0.13-1.fc13         fedora
```

Lists installed packages.

```
# yum lsi
...
zip.x86_64             3.0-1.fc13          @anaconda-InstallationRepo.x86_64
zlib.i686              1.2.3-23.fc12       @fedora/12
zlib.x86_64            1.2.3-23.fc12       @anaconda-
InstallationRepo.x86_64/12
zlib-devel.x86_64      1.2.3-23.fc12       @fedora/12
zvbi.x86_64            0.2.33-4.fc12       @fedora/12
```

Lists all available updates.

```
# yum lsu
...
curl.x86_64            7.20.1-4.fc13       updates
libcurl.i686           7.20.1-4.fc13       updates
libcurl.x86_64         7.20.1-4.fc13       updates
```

Updates a package.

```
# yum up curl
...
Dependencies Resolved

=====
Package                        Arch      Version      Repository    Size
=====
Updating:
```

Chapter 6. Yum extensions

```
curl                x86_64                7.20.1-4.fc13                updates                203 k
Updating for dependencies:
libcurl             i686                  7.20.1-4.fc13                updates                181 k
libcurl             x86_64                7.20.1-4.fc13                updates                175 k

Transaction Summary
=====
Upgrade             3 Package(s)

Total download size: 559 k
Is this ok [y/N]:
...
```

Available alias

There are many predefined alias in **/etc/yum/aliases.conf** you can use, and of course, you can add new ones. Using **yum alias** you can list the configured aliases, in this case you get:

```
# yum alias
Loaded plugins: aliases, filter-data, presto
Alias ALL = --enablerepo=development --enablerepo=updates-testing --enablerepo=fedora-source
--enablerepo=updates-source --enablerepo=development-source --enablerepo=updates-testing-
source --enablerepo=fedora-debuginfo --enablerepo=updates-debuginfo --enablerepo=development-
debuginfo --enablerepo=updates-testing-debuginfo
Alias ALLDBG = --enablerepo=fedora-debuginfo --enablerepo=updates-debuginfo --
enablerepo=development-debuginfo --enablerepo=updates-testing-debuginfo
Alias ALLSRC = --enablerepo=fedora-source --enablerepo=updates-source --
enablerepo=development-source --enablerepo=updates-testing-source
Alias DBG = --enablerepo=fedora-debuginfo --enablerepo=updates-debuginfo
Alias DEV = --enablerepo=development
Alias DEVDBG = --enablerepo=development-debuginfo
Alias DEVSRC = --enablerepo=development-source
Alias DUPS = --showduplicates
Alias FORCE = --skip-broken --disableexcludes=all
Alias SEC = --security
Alias SRC = --enablerepo=fedora-source --enablerepo=updates-source
Alias UPT = --enablerepo=updates-testing
Alias UPTDBG = --enablerepo=updates-testing-debuginfo
Alias UPTSRC = --enablerepo=updates-testing-source
Alias chk = check-update
Alias dbg-inst = --enablerepo=fedora-debuginfo --enablerepo=updates-debuginfo install
Alias dbg-install = --enablerepo=fedora-debuginfo --enablerepo=updates-debuginfo install
Alias down = downgrade
Alias force-inst = --skip-broken --disableexcludes=all install
Alias force-install = --skip-broken --disableexcludes=all install
Alias force-up = --skip-broken --disableexcludes=all upgrade
Alias force-update = --skip-broken --disableexcludes=all update
Alias force-upgrade = --skip-broken --disableexcludes=all upgrade
Alias h = history
Alias hi = history info
Alias hl = history list
Alias hs = history summary
Alias in = install
Alias inst = install
Alias ls = list
Alias lsa = list available
Alias lsi = list installed
Alias lssu = --security list updates
Alias lsu = list updates
Alias lupi = list-updateinfo
Alias rein = reinstall
Alias rm = remove
Alias up = upgrade
Alias v = version
alias done
```

Add an alias with yum alias

```
# yum alias ver version
...
alias done
```

```
# yum alias ver
...
Alias ver = version
alias done
```

6.3. yum-plugin-auto-update-debug-info

This extension checks if debuginfo packages are installed, and if so, it enables all the debuginfo repositories which are "children" of the enabled repositories to allow automatic updating of these packages.

6.3.1. Configuration

/etc/yum/pluginconf.d/auto-update-debuginfo.conf

```
[main]
enabled=1
```

enabled = 0|1

Enable extension at a general level.

6.3.2. Usage examples

The package *zaz-debuginfo* is already installed.

```
# yum info zaz-debuginfo
...
Loaded plugins: aliases, presto
Installed Packages
Name       : zaz-debuginfo
Arch       : x86_64
Version    : 0.3.3
Release    : 1.fc13
Size       : 3.9 M
Repo       : installed
From repo  : fedora-debuginfo
Summary    : Debug information for package zaz
URL        : http://sourceforge.net/projects/zaz/
License    : GPLv3+ and CC-BY-SA
Description: This package provides debug information for package zaz.
             Debug information is useful when developing applications that
             use this package or when debugging this package.
```

In normal conditions, ie, without this extension installed and enabled, no update is detected:

```
# yum check-update
...
ibus-chewing.x86_64      1.3.7.20100910-1.fc13      updates
psacct.x86_64           6.5.4-2.fc13              updates
yum.noarch              3.2.28-4.fc13             updates
```

Converting obsolete packages		
kwebkitpart.x86_64	0.9.6-1.fc13	updates
webkitpart.x86_64	0.0.5-0.2.svn1088283.fc13	@fedora
yum.noarch	3.2.28-4.fc13	updates
yum-plugin-download-order.noarch	0.2-1.fc11	@fedora

Installing and enabling **yum-plugin-auto-update-debug-info** the result changes:

```
# yum check-update

Loaded plugins: aliases, auto-update-debuginfo, presto
Found 1 installed debuginfo package(s)
...
ibus-chewing.x86_64                1.3.7.20100910-1.fc13      updates
psacct.x86_64                     6.5.4-2.fc13               updates
yum.noarch                         3.2.28-4.fc13              updates
zaz-debuginfo.x86_64              1.0.0-1.fc13               updates-debuginfo
Converting obsolete packages
kwebkitpart.x86_64                0.9.6-1.fc13               updates
webkitpart.x86_64                 0.0.5-0.2.svn1088283.fc13  @fedora
yum.noarch                        3.2.28-4.fc13              updates
yum-plugin-download-order.noarch  0.2-1.fc11                 @fedora
```

6.4. yum-plugin-changelog

This extension adds the **--changelog** option to the command line and it allows you to view package spreads changes (changelog) before or after updating packages.

6.4.1. Configuration

/etc/yum/pluginconf.d/changelog.conf

```
[main]
enabled=1
when=pre
always=false
```

enabled = 0|1

Enable extension at a general level.

when = pre|post

The changelog is printed before or after the update. It defaults to *pre*.

always = true|false

If it is set to *true*, it causes to always print the changelog, removing the option from the command line.

6.4.2. Usage examples

```
# yum update selinux-policy --changelog
Loaded plugins: aliases, changelog, filter-data, local, merge-conf, presto,
                : refresh-packagekit, tmprepo
Skipping filters plugin, no data
Setting up Update Process
Resolving Dependencies
```

```

Skipping filters plugin, no data
--> Running transaction check
--> Processing Dependency: selinux-policy = 3.9.7-16.fc14 for package: selinux-policy-targeted-3.9.7-16.fc14.noarch
--> Processing Dependency: selinux-policy = 3.9.7-16.fc14 for package: selinux-policy-targeted-3.9.7-16.fc14.noarch
---> Package selinux-policy.noarch 0:3.9.7-18.fc14 set to be updated
-> Running transaction check
---> Package selinux-policy-targeted.noarch 0:3.9.7-18.fc14 set to be updated
--> Finished Dependency Resolution

```

Changes in packages about to be updated:

```

ChangeLog for: selinux-policy-3.9.7-18.fc14.noarch,
               : selinux-policy-targeted-3.9.7-18.fc14.noarch
* Thu Dec 16 07:30:00 2010 Miroslav Grepl <mgrepl@redhat.com> 3.9.7-18
- Other fixes for munin plugins policy

* Wed Dec 15 07:30:00 2010 Miroslav Grepl <mgrepl@redhat.com> 3.9.7-17
- Fixes for sandbox policy
- Add setuid capability for vpnc
- Allow sandbox to run on nfs partitions
- Allow domains that transition to ping or traceroute, kill them
- Allow user_t to conditionally transition to ping_t and traceroute_t

```

Dependencies Resolved

```

=====
Package                        Arch      Version      Repository  Size
=====
Updating:
selinux-policy                 noarch    3.9.7-18.fc14  _local      744 k
Updating for dependencies:
selinux-policy-targeted        noarch    3.9.7-18.fc14  _local      2.4 M

```

Transaction Summary

```

=====
Upgrade      2 Package(s)

```

```

Total download size: 3.1 M
Is this ok [y/N]:
...

```

6.5. yum-plugin-downloadonly

This extension adds the **--downloadonly** option to allow to only download the packages without installing or updating.

6.5.1. Configuration

/etc/yum/pluginconf.d/downloadonly.conf

```

[main]
enabled=1

```

enabled = 0|1

Enable extension at a general level.

6.5.2. Usage examples

The following is an example of how to use this option with an uninstalled package.

```
# yum install --downloadonly 3proxy
Loaded plugins: aliases, downloadonly, filter-data, local, merge-conf,
               : presto, refresh-packagekit, tmprepo
Skipping filters plugin, no data
Setting up Install Process
Resolving Dependencies
Skipping filters plugin, no data
--> Running transaction check
--> Package 3proxy.x86_64 0:0.6.1-10.fc14 set to be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package           Arch           Version           Repository        Size
=====
Installing:
3proxy            x86_64         0.6.1-10.fc14     updates           150 k

Transaction Summary
=====
Install          1 Package(s)

Total download size: 150 k
Installed size: 495 k
Is this ok [y/N]: y
Downloading Packages:
Setting up and reading Presto delta metadata
Processing delta metadata
Package(s) data still to download: 150 k
3proxy-0.6.1-10.fc14.x86_64.rpm          | 150 kB      00:01

exiting because --downloadonly specified
```

Note now that **yum** ends differently. Now the question is, where is the rpm file downloaded? The general answer is that it depends on **yum** configuration, which defaults to **/var/cache/yum/<repositorio>** in Fedora. In the example, the downloaded rpm file comes from Fedora x86_64 [updates], the directory name where it ended is **/var/cache/yum/x86_64/13/updates/packages**.

```
# ls /var/cache/yum/x86_64/13/updates/packages/
3proxy-0.6.1-9.fc13.x86_64.rpm
```

The rpm downloaded is there, and you can confirm that it was not installed with :

```
# rpm -q 3proxy
package 3proxy is not installed
```

Usage with package updates.

```
# yum --downloadonly upgrade yum-utils
...
Downloading Packages:
(1/1): yum-utils-1.1.6-1. 100% |=====| 47 kB      00:02

exiting because --downloadonly specified
[root@dica ~]# ls /var/cache/yum/updates/packages/
yum-utils-1.1.6-1.fc7.noarch.rpm
```

6.5.3. What happens with the package dependencies?

```
# 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
```

yum does again the right thing, downloading everything but not installing.

```
# ls
...
zaptel-lib-1.4.0-9.fc6.i386.rpm
zaptel-utils-1.4.0-9.fc6.i386.rpm
```

So you may download the packages and its dependencies with this extension for a computer with no internet connection, for example.

6.5.4. What happens with packages already downloaded?

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

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

It makes sense, it's already installed. Is there a way to force it? No, there is not. There is another program to solve this situation, and it's documented below.

6.6. yum-plugin-fastestmirror

This extension is designed to list mirrors by response speed before a download begins.

When attempting an **yum** operation, there is always this sorting. You can check its effectiveness by disabling and enabling the extension doing the same download y denying the use of the cache.

6.6.1. Configuration

/etc/yum/pluginconf.d/fastestmirror.conf

```
[main]
enabled=1
verbose=0
always_print_best_host = true
socket_timeout=3
hostfilepath=timedhosts.txt
maxhostfileage=10
maxthreads=15
#exclude=.gov, facebook
#include_only=.nl, .de, .uk, .ie
```

enabled = 0|1

Enable extension at a general level.

verbose = true|false

Prints more information on the progress of computing.

always_print_best_host = true|false

Prints information about the best mirror after probing.

hostfilepath = ruta_archivo

Absolute or relative path to the file with the information of all measured mirrors, typically timedhosts.txt. If this is a relative path, cachedir is used as the root to the path.

maxhostfileage = integer

Max cache file age in days.

maxthreads = integer

Max number of threads used by the extension for its computings.

socket_timeout = integer

Timeout in seconds for the network connection to the repository being tested.

6.6.2. Usage examples

In this example, we will make a computation between using and not using the extension to determine its effectiveness.

The test consists of downloading all the metadata to make an extensive listing, but first we must make a cache cleanup.

```
# yum clean all
...
Cleaning repos: _local fedora fedora-chromium livna rpmfusion-free
                : rpmfusion-free-updates rpmfusion-nonfree
                : rpmfusion-nonfree-updates updates
Cleaning up Everything
507 delta-package files removed, by presto
```

fastestmirror must be enabled in the configuration file and then starts the measuring of time required to make a comprehensive list:


```
# time yum list available
...
zzuf.x86_64                                0.13-1.fc13                fedora

real    3m19.740s
user    0m16.017s
sys     0m1.269s
```

Now the experiment will be repeated disabling **fastestmirror**, but first, a cache cleanup is made.

```
# yum clean all
```

```
# time yum list --disableplugin=fastestmirror available
...
zziplib-utils.x86_64                    0.13.49-8.fc12            fedora
zzuf.x86_64                             0.13-1.fc13               fedora

real    3m24.678s
user    0m16.125s
sys     0m1.270s
```

The gain has been small. The details of how it measures the speed of the repositories is not known, but the truth is that it does not seem to be very effective. Even though, according to the experiments, it works according to its own logic, ie measuring all the servers and using the fastest according to that measure, hence there is potential for doing it better. One of the tests was to check whether this extension acts after assembling the list of servers regardless of the *failovermethod* method selected in the repositories configuration, ie, it works regardless of the order of that list, re-ordering it according to the speed measured and choosing the "fastest".

6.7. yum-plugin-filter-data

This extension adds **--filter-vendors**, **--filter-groups**, **--filter-packagers**, **--filter-licenses**, **--filter-arches**, **--filter-committers**, **--filter-buildhosts**, **--filter-baseurls**, **--filter-package-sizes**, **--filter-archive-sizes** and **--filter-installed-sizes** options to the command line. Note that each packet must match at least one pattern/range in each category, if any is specified.

6.7.1. Configuration

/etc/yum/pluginconf.d/filter-data.conf

```
[main]
enabled=1
```

enabled = 0|1

Enable extension at a general level.

6.7.2. Documentation

This extension includes an online manual page, see details with **man yum-filter-data**.

6.7.3. Usage examples

List updates with package sizes less than 1MB:

```
$ yum --filter-package-sizes=-1m check-update
Loaded plugins: aliases, filter-data, presto, refresh-packagekit
_local | 3.0 kB 00:00
_local/primary_db | 306 kB 00:00
fedora | 4.3 kB 00:00
fedora/primary_db | 14 MB 01:58
fedora-chromium | 3.0 kB 00:00
fedora-chromium/primary_db | 12 kB 00:00
updates | 4.7 kB 00:00
updates/primary_db | 3.3 MB 00:29
Limiting package lists to filtered ones
Left with 1 of 2 packages, after filters applied
selinux-policy.noarch 3.9.7-18.fc14 _local
selinux-policy-targeted.noarch 3.9.7-18.fc14 updates
```

Apply updates Tim Waugh has provided:

```
$ yum --filter-committers="Tim Waugh *" update
...
Limiting package lists to filtered ones
Left with 6 (+0 related) of 18 packages, after filters applied
...
Dependencies Resolved

=====
Package Arch Version Repository Size
=====
Updating:
system-config-printer x86_64 1.1.16-8.fc12 updates 434 k
system-config-printer-libs x86_64 1.1.16-8.fc12 updates 643 k
system-config-printer-udev x86_64 1.1.16-8.fc12 updates 55 k

Transaction Summary
=====
Install 0 Package(s)
Update 3 Package(s)

Total download size: 1.1 M
Is this ok [y/N] : y
```

List updates for a specific group:

```
$ yum --filter-rpm-groups="App*/Internet" list updates
...
Limiting package lists to filtered ones
Left with 1 of 9 packages, after filters applied
Updated Packages
purple-facebookchat.x86_64 1.64-1.fc12 updates
```

Upgrade a specific set of rpm groups:

```
$ yum --filter-rpm-groups="App*/Internet,Dev*/Lib*" update
...
Limiting package lists to filtered ones
Left with 4 (+0 related) of 18 packages, after filters applied
...
Dependencies Resolved

=====
Package Arch Version Repository Size
=====
Updating:
libxcb-devel x86_64 1.5-1.fc12 updates 139 k
```

```

purple-facebookchat      x86_64      1.64-1.fc12      updates      47 k
Updating for dependencies:
libxcb                   x86_64      1.5-1.fc12       updates      99 k

Transaction Summary
=====
Install      0 Package(s)
Update       3 Package(s)

Total download size: 285 k
Is this ok [y/N]:

```

List updates for a specific yum group:

```

$ yum --filter-groups="Development Tools,Server Configuration Tools" list updates
...
Limiting package lists to filtered ones
Left with 2 of 9 packages, after filters applied
Updated Packages
gdb.x86_64                        7.0.1-26.fc12      updates
system-config-printer.x86_64     1.1.16-8.fc12      updates

```

Update a specific set of yum groups:

```

$ yum --filter-groups="X Window System,Printing Support" update
...
Limiting package lists to filtered ones
Left with 4 (+0 related) of 18 packages, after filters applied
...
Dependencies Resolved

=====
Package                Arch    Version      Repository Size
=====
Updating:
system-config-printer   x86_64   1.1.16-8.fc12 updates   434 k
system-config-printer-udev x86_64  1.1.16-8.fc12 updates    55 k
Updating for dependencies:
system-config-printer-libs x86_64  1.1.16-8.fc12 updates   643 k

Transaction Summary
=====
Install      0 Package(s)
Update       3 Package(s)

Total download size: 1.1 M
Is this ok [y/N]:

```

6.8. yum-plugin-fs-snapshot

yum-fs-snapshot is an extension to make backups of your file systems before executing a **yum** transaction. By default, it will make an image (snapshot) of any file system that is able, for now limited to btrfs file system. However, all file systems on LVM logical volumes can be used to block-level images using LVM. The LVM images support is provided in order to make a turning back of the system (rollback). Note that these LVM images will only be created if the kernel supports DM "snapshot-merge".

6.8.1. Configuration

`/etc/yum/pluginconf.d/fs-snapshot.conf`

```
[main]
enabled = 1

[lvm]
enabled = 0
# 'lvcreate_size_args' option must specify the snapshot LV size using -L or -l
#lvcreate_size_args = -l 15%ORIGIN
```

enabled = 0|1

Enable extension at a general level.

6.8.2. Documentation

```
# rpm -qd yum-plugin-fs-snapshot
/usr/share/man/man1/yum-fs-snapshot.1.gz
/usr/share/man/man5/yum-fs-snapshot.conf.5.gz
```

6.9. yum-plugin-keys

This extension adds the commands **keys**, **keys-info**, **keys-data** and **keys-remove**. These commands allow to query and remove signature keys.

yum keys [key-wildcard]

List keys to sign data, alias: **keys-list**.

yum keys-info [key-wildcard]

Complete information about data signing keys.

yum keys-data [key-wildcard]

Shows the public block of information of the data signing key.

yum keys-remove [key-wildcard]

Removes a public key for signing data, alias: **keys-erase**.

Currently it's not possible to add a gpg key with **yum** but this is not a problem in most cases because when you install a signed package and settings point to a key, **yum install** allows to import the key when necessary.

6.9.1. Configuration

/etc/yum/pluginconf.d/keys.conf.

```
[main]
enabled=1
```

enabled = 0|1

Enable extension at a general level.

6.9.2. Usage examples

Key listing

```
# yum keys
Loaded plugins: aliases, changelog, dellsysidplugin2, fastestmirror, filter-data, keys, list-
data, presto, refresh-packagekit, verify
Loading mirror speeds from cached hostfile
Key owner                                Key ID                                Key email
Repo
Adobe Systems Incorporated (Linux RPM Signing Key)      secure@adobe.com
installed                                              f6777c67-45e5b1b9
Fedora (10 testing)                                     fedora@fedoraproject.org
installed                                             0b86274e-48b5dd6f
Fedora (10)                                              fedora@fedoraproject.org
installed                                             4ebfc273-48b5dbf3
Fedora (11)                                              fedora@fedoraproject.org
installed                                             d22e77f2-49ee1409
Fedora (12)                                              fedora@fedoraproject.org
installed                                             57bbccba-4a6f97af
Fedora (8 and 9)                                        fedora@fedoraproject.org
installed                                             6df2196f-48b5d596
Fedora Project                                         fedora@redhat.com
installed                                             4f2a6fd2-3f9d9d3b
Google, Inc. Linux Package Signing Key                linux-packages-
keymaster@google.com                                installed                                7fac5991-4615767f
Livna.org rpms                                         rpm-key@livna.org
installed                                             a109b1ec-3f6e28d5
Michael Fleming (RPM key) (RPM Signing Key)            mfleming
+rpm@enlartenment.com                                installed                                536ceb2a-419e7fcd
PPTP Software Signing Key                             pptp@city-fan.org
installed                                             862acc42-42243bfc
RPM Fusion free repository for Fedora (11)             rpmfusion-
buildsys@lists.rpmfusion.org                         installed                                8fcff4da-49c51006
RPM Fusion free repository for Fedora (12)             rpmfusion-
buildsys@lists.rpmfusion.org                         installed                                16ca1a56-4a100959
RPM Fusion nonfree repository for Fedora (11)          rpmfusion-
buildsys@lists.rpmfusion.org                         installed                                8dc43844-49c510d6
RPM Fusion nonfree repository for Fedora (12)          rpmfusion-
buildsys@lists.rpmfusion.org                         installed                                a3a882c1-4a1009ef
RPM Fusion repository (Fedora - free)                  rpmfusion-
buildsys@lists.rpmfusion.org                         installed                                49c8885a-4878ddfb
RPM Fusion repository (Fedora - nonfree)               rpmfusion-
buildsys@lists.rpmfusion.org                         installed                                b1981b68-4878de85
keys done
```

Specific information of a key

```
# yum keys-data 4ebfc273-48b5dbf3
Loaded plugins: aliases, changelog, dellsysidplugin2, fastestmirror, filter-data, keys, list-
data, presto, refresh-packagekit, verify
Loading mirror speeds from cached hostfile
Type      : GPG
Key owner : Fedora (10)
Key email : fedora@fedoraproject.org
Key ID    : 4ebfc273
Repo      : installed
Created   : Wed Aug 27 18:27:55 2008
Raw Data :
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: rpm-4.6.0-rc1 (NSS-3)

mQGibEi12/MRBADL6fpbRejYViiVzY/Y0T6pmgCh0ykrsq1oc9mX9+ss1lZZit1z
DRrgvOYG67wnRQaskXnKu1w3XoNEAT/OjCNzDwrS3gMbHHz+T9wQN5We052ihozp
ZLo/aNnIW9a7W5KxMyrmDobo5919wjiHXFXbVvH5kgm1B36ZG09/HlwtpwCgyW/Z
qcZIs990hIrwYZRUJ8BHNdED/R6SFCF2p4AqMkpo6b09+N5BLvTRPythuWyfTpOa
XGLPDzkDisIW72DDmCrVI4RQnLW5u0xko7L1o8zw0mmFg6C8HLeeHpfg40MVLDDJ
OtX1UTPMKms6eqKYWn15266y8Bga5QnVNGMRPXK0YJhog8K4h7MV5i/B7F2Byhu
rcT2A/42Ikj3qoA7mxIEgu3DDrxboKWZghAIWamN8xqIl0/fIMmx/x3k6lDakQRq
```

```
+E6kdTvrWxsKbu5ESTzCDIjhHyVIECLGJ+EbB+etIa3N1+ek3gKEBcDHBtvbdc57
NcYCWvDvL30/SNpkKetm4kzrvAqL0Q6mvmGjSY4odoIPczhS1bQmRmVkb3JhICgx
MCKgPGZLZG9YyUBmZWRvcmFwcm9qZWNOm9yZz6IYAQTEQIAUAUSLXb8wIbAwYL
CQgHAWIEFQIIAwQWAgMBAh4BAheAAoJEL8ib8x0v8JzaaEAnjrJTa3QCZr9C91r
MYJ2uNQvMQ5rAKCfGK2Q2AVxK3pK1QHtc972g5gTrkEDQRitdvzEBAAz4E9yA4G
epLon5F70Ak4/OgpJgHvXFN2xMGazpExkh+3/bjil3/J6rucXtLutPP4kIIu6LgW
sbDF3HUafwpeOQI7Yx1MwLXTfZ0Mevrsm4wEBSYoC8U4zygVyz2m6Rz2i62hbj/W
AYIayW2sXg7VwbFoJxjjZfw2d6jTNQW9EI3GVffYUN+V7BSeQw7Y6v18pxFUXCur
qOGXwNSUduUmdh1h20+CmvdSiYJApEU/fBHNioTDNVynTVN1ayobTs/5xANnpG+K
7Lhiz2HrY6tTv34DaDMwv2aMoDxq5BTfi+H7rTzXlpF5fAjHNGsXQIJjKhEgpkHo
hr+in0s6CsDpERpleS8FKihjNwcGksoviVpEitnsHTOV0y0V5ZiuZpZ8M6ZS4Vf/
JhvRCK2CeCkeyaAgxsGngN/QeTHE4e35m/dqUnBrUKRNkpcJ8jJ011eVNj0Xe/EG
Vcy345TiUCYG6kr1wJjt0xJm1wDricUhnR2X5motCOT/k6BF0ucofvcy10u62+N
DKgIOE/o3BMo9XfPJvteheM0hWRMfNwmbFdr13rKk/GZyVeZmLw1Fx1byH5r4J5T
4vE9JmgeM2bVu/TiMfLbszQM39xcE3SycjX5knBEglzad23kzyIerlv8RZUdQJ/L
qfmRXc+0x0H565YLwGC79XCKUp7d1GBYhL8ABAsP/1DrF/Szknki1KwMeK8DP34q
GBDwtUUVTiMqHZclUyLHRT3bp2Ae2/+xy7NAy/Y/G5aZqkAMvaqiG3ps5LDVJo+1
WAQB0gnwrZtVn6FU6H59a4h79hKoFoBEI1gd9XKzzgA+RF4cEHJiuFqh2boxdr7/
n64tZrS6HkUcD15wIcYoYl6i0HBFbazbyZifeKMu0USkrfzLRLPec2V3p9bXqg1W
h+SBCpD9uhSzBeKs636ipOL3WZ9UUMQMPNKyyFxx9cKcZY9aP0Leub2kvnFIZiG7
ljbRP1dNj1himTxTjZzf0icxu5MaSTyIl9ZVEPX0mPwi/a8POP82pLZYZDi8Scx
zQFNi+22VdSrQkj6Jkan3GoqB0BoSv9iQ7EJp7wpmBRhoacW3wtJ4m0+cAR8IX9X
6DTklAU9zoZzdvg063GnjgPCi30hw12jvM1NUcn4vdI5UUTd7DAPaysueJPqfFCg
CSgr0cCyfQL4vL0iRp82NNQtUS9ZdyUw8t6w/GuFnsymBSTqt2t1n+mhS7RAejaV
1QV/KmwijbaVh2LjTP0YzPWZcBy8XHz0snuWufFmRTgFOUmWT9o82RSde6ubvnDN
WfWwGZriIpJd3JCBpkVb2aiEkU4958Ai36I/splnx0rAJZV/4ZIIHnGMwn23/DNh
gX4TCgPMjjjT0hmTfDpliEkEBECAAKFAki12/MCGwwACgkQvyJvzE6/wnM+4wCe
KkKsIfsmCT0mpuYxHh4vRS/ix58AoLfjES8VpN+3ku9hC3QT97Fa/bK4
=s6ix
-----END PGP PUBLIC KEY BLOCK-----
```

keys-data done

Key removal

```
# yum keys-remove a7a1d4b6-43281558
Loaded plugins: aliases, filter-data, keys, local, merge-conf, presto,
               : refresh-packagekit, tmprepo
Resolving Dependencies
--> Running transaction check
---> Package gpg-pubkey.noarch 0:a7a1d4b6-43281558 set to be erased
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package           Arch      Version                      Repository      Size
=====
Removing:
gpg-pubkey        noarch    a7a1d4b6-43281558          installed      0.0

Transaction Summary
=====
Remove           1 Package(s)

Installed size: 0
Is this ok [y/N]: y
Downloading Packages:
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
Erasing          : gpg-pubkey-a7a1d4b6-43281558.noarch      1/1

Removed:
```

```
gpg-pubkey.noarch 0:a7a1d4b6-43281558
```

```
Complete!
```

6.10. yum-plugin-list-data

This extension add the commands:

- **list-vendors**
- **info-vendors**
- **list-rpm-groups**
- **info-rpm-groups**
- **list-packagers**
- **info-packagers**
- **list-licenses**
- **info-licenses**
- **list-arches**
- **info-arches**
- **list-committers**
- **info-committers**
- **list-buildhosts**
- **info-buildhosts**
- **list-baseurls**
- **info-baseurls**
- **list-package-sizes**
- **info-package-sizes**
- **list-archive-sizes**
- **info-archive-sizes**
- **list-installed-sizes**
- **info-installed-sizes**
- **list-groups**
- **info-groups**

They all take the same arguments as the command **list** and **info**. The difference between **list** and **info** variants is that **info** lists all packages under each addition.

Example of syntax:

yum list-vendors [PACKAGE|all|installed|updates|extras|obsoletes|recent]

Shows the data added in vendor attribute of a package group.

yum info-vendors [PACKAGE|all|installed|updates|extras|obsoletes|recent]

Shows the data added in vendor attribute of a package group and lists all packages in each one.

6.10.1. Documentation

This extension includes an online manual page, see `man yum-list-data`.

6.10.2. Configuration

`/etc/yum/pluginconf.d/list-data.conf`

```
[main]
enabled=1
```

6.10.3. Usage examples

Finding out the rpm group of a package

```
$ yum list-rpm-groups yum-list-data
...
===== Installed Packages =====
System Environment/Base      1 (100%)
===== Available Packages =====
System Environment/Base      1 (100%)
list-rpm-groups done
```

Package sizes when installed

```
# yum info-installed-sizes
...
[ 1B - 10KB ]      886 ( 6%)
Allegro0GG-devel-1.0.3-4.fc9.i386      6.2 k ( 6,363)
Frodo-gui-4.1b-5.fc10.i386             8.8 k ( 9,038)
Io-language-mysql-20071010-6.fc10.i386  9.8 k (10,064)
Io-language-postgresql-20071010-6.fc10.i386  5.0 k ( 5,164)
...
# yum info-installed-sizes firefox
...
===== Installed Packages =====
-- Unknown --      1 (100%)
firefox-3.0.5-1.fc10.i386

===== Available Packages =====
[ 10MB - 50MB ]      2 (100%)
firefox-3.0.5-1.fc10.i386      14 M (15,028,447)
firefox-3.0.4-1.fc10.i386      14 M (14,831,870)

info-installed-sizes done
# yum info-installed-sizes firefox-3.0.5
===== Installed Packages =====
-- Unknown --      1 (100%)
firefox-3.0.5-1.fc10.i386

===== Available Packages =====
```



```
[ 10MB - 50MB ]      1 (100%)
firefox-3.0.5-1.fc10.i386      14 M (15,028,447)

info-installed-sizes done
```

6.11. yum-plugin-local

When this extension is installed and enabled, it will automatically copy all installed packages to a repository on the local file system and rebuild the repository. This means that any package that you downloaded and installed with yum **yum** will always exist, even if it is removed from the original repository, therefore, it can be uninstalled, reinstalled, downgraded, etc.

6.11.1. Configuration

The provided configuration files are:

- `/etc/yum/repos.d/_local.repo`
- `/etc/yum/pluginconf.d/local.conf`

`/etc/yum/pluginconf.d/local.conf`

```
[main]
enabled=1

# Createrepo options. See man createrepo
[createrepo]
# If you want to speedup createrepo with the --cachedir option. Eg.
# cachedir = /tmp/createrepo-local-plugin-cachedir
# If you need to override the checksum with the --checksum option. Eg.
# checksum = sha

quiet = true
verbose = false
skip_stat = false
unique_md_filenames = true
update = true
databases = true
```

enabled = 0|1

Enable extension at a general level.

quiet = true|false

Silent.

verbose = true|false

Report many details.

skip_stat = true|false

Avoid file/directory verification with stat.

unique_md_filenames = true|false

Adds file checksum in metadata file name.

update = true|false

If metadata exists in the directory and the rpm has not changed since metadata was created, it will reuse these metadata when set to true instead of recreate them.

`databases = true|false`

It creates the sqlite **yum** database.

`/etc/yum.repos.d/_local.repo`

When this extension is enabled, this file is automatically created when you install a new package. Once created, it can be customized if necessary.

```
[_local]
name=Automatic local repo. (managed by the "local" yum plugin).
baseurl=file:/var/lib/yum/plugins/local
enabled=1
gpgcheck=true
# Metadata expire could be set to "never" because the local plugin will
# automatically cause a cache refresh when new packages are added. However
# it's really cheap to check, and this way people can dump stuff in whenever
# and it never gets out of sync. for long.
metadata_expire=1h
# Make cost smaller, as we know it's "local". If you really want to be sure,
# you can do this ... but the name will do pretty much the same thing, and that
# way we can also see the other packages (with: --showduplicates list).
# cost=500
```

6.11.2. Usage examples

Create local repo with **yum install**

```
# yum install zziplib-devel
...
Resolving Dependencies
Skipping filters plugin, no data
--> Running transaction check
--> Package zziplib-devel.x86_64 0:0.13.49-8.fc12 set to be installed
--> Processing Dependency: zziplib = 0.13.49-8.fc12 for package: zziplib-
devel-0.13.49-8.fc12.x86_64
--> Processing Dependency: SDL-devel for package: zziplib-devel-0.13.49-8.fc12.x86_64
--> Processing Dependency: libzipfseeko-0.so.13()(64bit) for package: zziplib-
devel-0.13.49-8.fc12.x86_64
--> Processing Dependency: libzip-0.so.13()(64bit) for package: zziplib-
devel-0.13.49-8.fc12.x86_64
--> Processing Dependency: libzipmapped-0.so.13()(64bit) for package: zziplib-
devel-0.13.49-8.fc12.x86_64
--> Processing Dependency: libzipwrap-0.so.13()(64bit) for package: zziplib-
devel-0.13.49-8.fc12.x86_64
--> Running transaction check
--> Package SDL-devel.x86_64 0:1.2.14-8.fc14 set to be installed
--> Package zziplib.x86_64 0:0.13.49-8.fc12 set to be installed
--> Finished Dependency Resolution
```

Dependencies Resolved

Package	Repository	Arch	Size	Version
Installing:				
zzilib-devel	fedora	x86_64	126 k	0.13.49-8.fc12
Installing for dependencies:				
SDL-devel	fedora	x86_64	362 k	1.2.14-8.fc14

```

zziplib                                x86_64                                0.13.49-8.fc12
                                fedora                                64 k

Transaction Summary
=====
Install      3 Package(s)

Total download size: 552 k
Installed size: 1.9 M
Is this ok [y/N]: y
Downloading Packages:
Setting up and reading Presto delta metadata
Processing delta metadata
Package(s) data still to download: 552 k
(1/3): SDL-devel-1.2.14-8.fc14.x86_64.rpm
                                | 362 kB      00:03
(2/3): zziplib-0.13.49-8.fc12.x86_64.rpm
                                |  64 kB      00:00
(3/3): zziplib-devel-0.13.49-8.fc12.x86_64.rpm
                                | 126 kB      00:01
-----
Total
                                66 kB/s | 552 kB      00:08

Previous repo file missing:
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing      : zziplib-0.13.49-8.fc12.x86_64
                                1/3
  Installing      : SDL-devel-1.2.14-8.fc14.x86_64
                                2/3
  Installing      : zziplib-devel-0.13.49-8.fc12.x86_64
                                3/3

Installed:
  zziplib-devel.x86_64 0:0.13.49-8.fc12

Dependency Installed:
  SDL-devel.x86_64 0:1.2.14-8.fc14
  zziplib.x86_64 0:0.13.49-8.fc12

Complete!

```

Removal and reinstallation

```

# yum remove SDL-devel
...
Running Transaction
  Erasing      : zziplib-devel-0.13.49-8.fc12.x86_64
                                1/2
  Erasing      : SDL-devel-1.2.14-8.fc13.x86_64
                                2/2

Erased:
  SDL-devel.x86_64 0:1.2.14-8.fc13

Dependencie(s) erased:
  zziplib-devel.x86_64 0:0.13.49-8.fc12

Complete!

```

```

# yum install SDL-devel
Loaded plugins:aliases, appmarket, auto-update-debuginfo,

```

```

: downloadonly, filter-data, keys, list-data, local,
: presto, verify
_local | 3.0 kB 00:00 ...
_local/primary_db | 4.0 kB 00:00 ...
rubyera | 3.2 kB 00:00 ...
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package SDL-devel.x86_64 0:1.2.14-8.fc13 set to be installed

Dependencies Resolved

=====
Package Arch Version Repository Size
=====
Installing:
SDL-devel x86_64 1.2.14-8.fc13 _local 362 k

Transaction Summary
=====
Install 1 Package(s)

Total download size: 362 k
Is this ok [y/N]:

```

Note that you do not need network connectivity to access the previously installed packages and that the repository in use is *_local*. Also note that this local repository will have all packages installed without distinction of its source repository, it is not a mirror of a particular repository and you can clean your cache without affecting the contents of this repository.

6.12. yum-plugin-merge-conf

This extension adds the command line option **--merge-conf**. With this option **yum** will ask what to do with the configuration files changed with a package update.

Available options are:

diff the two versions (d)

View the differences and go back to options menu (press q).

do the default RPM action (q)

Do the default RPM action, the new configuration file are installed with *.rpmnew* suffix, and current configuration files are not modified.

install the package's version (i)

Rename current configuration files to *.rpmsave* and installs the new configuration files provided by the package.

merge interactively with vim (v)

Allows you to interactively merge files using **vim**.

background this process and examine manually (z)

Sets the installation process in the background and lets you edit both files and manually resolve the differences. It gets to the installation process and present again the same last options to make a decision.

To go back to the installation process execute **exit** command.

6.12.1. Configuration

`/etc/yum/pluginconf.d/merge-conf.conf`

```
[main]
enabled=1

# Set to true, to always get the output (removes the cmd line arg)
always=false
```

`enabled = 0|1`

Enable the extension at system wide.

`always = true|false`

Removes the command line argument and always ask what to do with the configuration files that have changed.

6.12.2. Usage examples

In this example we will call **vim** to solve arguments merging.

```
# yum --merge-conf update squid
...
Dependencias resueltas
=====
Paquete          Arquitectura      Versión              Repositorio      Tamaño
=====
Actualizando:
squid             x86_64            7:3.1.8-1.fc13      updates          1.6 M

Resumen de la transacción
=====
Upgrade          1 Package(s)

Tamaño total de la descarga: 1.6 M
Está de acuerdo [s/N]:
...
Ejecutando transacción
Actualizando    : 7:squid-3.1.8-1.fc13.x86_64                1/2
warning: /etc/squid/squid.conf created as /etc/squid/squid.conf.rpmnew
Limpieza       : 7:squid-3.1.1-4.fc13.x86_64                2/2

Package squid: merging configuration for file "/etc/squid/squid.conf":
By default, RPM would keep your local version and rename the new one to /etc/squid/
squid.conf.rpmnew
What do you want to do ?
- diff the two versions (d)
- do the default RPM action (q)
- install the package's version (i)
- merge interactively with vim (v)
- background this process and examine manually (z)
Your answer ?  v
```

```
#
visible hostname "movix.fedora-ve.org"
# Recommended minimum configuration:
#
acl manager proto cache_object
acl localhost src 127.0.0.1/32 ::1
acl to_localhost dst 127.0.0.0/8 0.0.0.0/32 ::1

# Example rule allowing access from your local n
# Adapt to list your (internal) IP networks from
# should be allowed
acl localnet src 10.0.0.0/8 # RFC1918 possib
acl localnet src 172.16.0.0/12 # RFC1918 possib
acl localnet src 192.168.0.0/16 # RFC1918 possib
acl localnet src fc00::/7 # RFC 4193 local
acl localnet src fe80::/10 # RFC 4291 link-

#
# Recommended minimum Access Permission configur
#
# Only allow cachemgr access from localhost
http_access allow manager localhost
http_access deny manager

# Deny requests to certain unsafe ports

squid.conf 3,2 Comienzo squid.conf.rpmnew
```

The call for merging is done using **vimdiff** application which allows to view clearly the differences.

6.13. yum-plugin-post-transaction-actions

yum extension to execute arbitrary commands when acting on certain packages. This extension allows the user to execute arbitrary actions immediately after a transaction over specific packages.

6.13.1. Configuration

/etc/yum/pluginconf.d/post-transaction-actions.conf

```
[main]
enabled = 1
actiondir = /etc/yum/post-actions/
```

enabled = 0|1

Enable the extension at system wide.

actiondir

Path to search for files with specific actions.

6.13.2. Usage examples

/usr/share/doc/yum-plugin-post-transaction-actions-1.1.24/sample.action

```
#action_key:transaction_state:command
# action_key can be: pkgglob, /path/to/file (wildcards allowed)
# transaction_state can be: install,update,remove,any
# command can be: any shell command
# the following variables are allowed to be passed to any command:
# $name - package name
# $arch - package arch
# $ver - package version
# $rel - package release
# $epoch - package epoch
# $repoid - package repository id
# $state - text string of state of the package in the transaction set
#
# file matches cannot be used with removes b/c we don't have the info available
```

```

*:install:touch /tmp/$name-installed
zsh:remove:touch /tmp/zsh-removed
zsh:install:touch /tmp/zsh-installed-also
/bin/z*h:install:touch /tmp/bin-zsh-installed
z*h:any:touch /tmp/bin-zsh-any

```

If you copy this sample file in `/etc/yum/post-actions/`, when you install **zsh** you get the next output:

```

# yum install zsh
...
Warning: Since last transaction, database has been modified (RPMDB).
Installing      : zsh-4.3.10-4.fc12.x86_64

...
Running post transaction command: touch /tmp/bin-zsh-any
Running post transaction command: touch /tmp/zsh-installed
Running post transaction command: touch /tmp/bin-zsh-installed
Running post transaction command: touch /tmp/zsh-installed-also
...

```

And when removing **zsh**:

```

# yum remove zsh
...
Warning: Since last transaction, databases has been modified (RPMDB).
Erasing       : zsh-4.3.10-4.fc12.x86_64

...
Running post transaction command: touch /tmp/bin-zsh-any
Running post transaction command: touch /tmp/zsh-removed
...

```

6.14. yum-plugin-priorities

This extension allows repositories to have different priorities. Packages in a low priority repository can not replace packages installed from repositories with higher priority, even if it has an updated version.

6.14.1. Configuration

`/etc/yum/pluginconf.d/priorities.conf`

```

[main]
enabled = 1
check_obsoletes=1

```

`enabled = 0|1`

Enable the extension at system wide.

`check_obsoletes=1|0`

If you do not want packages from low priority repos with the obsoletes specification to another one with a different name, set this option to 1.

Then each `.repo` file in `/etc/yum.repos.d/` may include `priority=N` to indicate the priority, where *N* is an integer between 1 and 99. The value 1 has the highest priority and 99 the lowest. By default the repository inherits a low priority with *N*=99.

6.14.2. Usage examples

In the following configuration the highest priority is given to the Fedora base repositories, which is highly recommended when using third party repositories.

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

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

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

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

6.15. yum-plugin-protectbase

This extension allows certain repositories to be protected. Packages in protected repositories can not be overridden by packages in non-protected repositories even if the unprotected repository has a newer version.

6.15.1. Configuration

/etc/yum/pluginconf.d/protectbase.conf

```
[main]
enabled = 1
```

enabled = 0|1

Enable the extension at system wide.

Later you need to configure the .repo files for each repository to declare which will be protected or not adding *protect=1* or *protect=0*.

6.15.2. Usage examples

In this example, packages from fedora-updates and fedora are protected against upgrades available for these packages from unprotected repositories.

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


```
[fedora]
name=Fedora $releasever - $basearch
failovermethod=priority
baseurl=http://download.fedoraproject.org/pub/fedora/linux/releases/$releasever/Everything/
$basearch/os/
mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=fedora-$releasever&arch=$basearch
enabled=1
metadata_expire=7d
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora-$basearch
protect=1
```

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

```
[updates]
name=Fedora $releasever - $basearch - Updates
failovermethod=priority
baseurl=http://download.fedoraproject.org/pub/fedora/linux/updates/$releasever/$basearch/
mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=updates-released-f
$releasever&arch=$basearch
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora-$basearch
protect=1
```

This configuration is highly recommended to those who like to add third party repositories, because it will protect them from breaking the Fedora base and will guarantee the update to new official versions.

6.16. yum-plugin-refresh-updatesd

yum-refresh-updatesd tells **yum-updatesd** to check the available updates when **yum** ends. This way if you run **yum list updates**, for example, and a new version for the **zsh** is available, **gpk-update-icon** will instantly be updated to reflect this.

6.16.1. Configuration

/etc/yum/pluginconf.d/refresh-updatesd.conf

```
[main]
enabled=1
```

enabled = 0|1

Enable extension at a general level.

6.17. yum-plugin-remove-with-leaves

This extension removes unused dependencies which have been incorporated by an installation package that would otherwise not be removed. Helps maintain a system clean of libraries and unused packages.

This extension adds the following options to the **remove** command:

--leaves-exclude-bin

Do not remove leaf packages containing executable binaries.

--remove-leaves

Removes unneeded dependencies for other packages.

6.17.1. Configuration

/etc/yum/pluginconf.d/remove-with-leaves.conf

```
[main]
enabled = 1
#exclude_bin = 1
#remove_always = 1
```

enabled = 0|1

Enable extension at a general level.

exclude_bin = 1|0

Do not remove leaf packages with binaries.

remove_always = 1|0

Always remove leaf packages.

6.17.2. Usage examples

```
$ yum remove --remove-leaves couchdb
...
Setting up Remove Process
Resolving Dependencies
--> Running transaction check
---> Package couchdb.i586 0:0.10.0-1.fc11 set to be erased
--> Finished Dependency Resolution
removing erlang-R12B-5.7.fc11.i586. It is not required by anything else.
removing libicu-devel-4.0.1-5.fc11.i586. It is not required by anything else.
removing js-1.70-8.fc11.i586. It is not required by anything else.
removing unixODBC-2.2.14-2.fc11.i586. It is not required by anything else.
--> Running transaction check
---> Package erlang.i586 0:R12B-5.7.fc11 set to be erased
---> Package js.i586 0:1.70-8.fc11 set to be erased
---> Package libicu-devel.i586 0:4.0.1-5.fc11 set to be erased
---> Package unixODBC.i586 0:2.2.14-2.fc11 set to be erased
--> Finished Dependency Resolution
Dependencies Resolved

=====
Package           Arch      Version      Repository      Size
=====
Removing:
couchdb           i586      0.10.0-1.fc11 installed        1.4 M
erlang            i586      R12B-5.7.fc11 installed         89 M
js                i586      1.70-8.fc11  installed        884 k
libicu-devel      i586      4.0.1-5.fc11 installed        3.0 M
unixODBC          i586      2.2.14-2.fc11 installed        1.1 M

Transaction Summary
=====
Remove           5 Package(s)
Reinstall        0 Package(s)
Downgrade        0 Package(s)

Is this ok [y/N]:
```

Note that this list of packages is exactly the same as when you install and no dependencies are preinstalled. Now compare it with the regular **yum remove** process:

```
$ yum remove couchdb
...
Setting up Remove Process
Resolving Dependencies
--> Running transaction check
---> Package couchdb.i586 0:0.10.0-1.fc11 set to be erased
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package           Arch      Version      Repository      Size
=====
Removing:
couchdb           i586      0.10.0-1.fc11 installed        1.4 M

Transaction Summary
=====
Remove           1 Package(s)
Reinstall        0 Package(s)
Downgrade        0 Package(s)

Is this ok [y/N]:
```

6.18. yum-plugin-warm-cache

This extension reads the rpmdb files and stores them in the system cache instead of accessing PMDB directly. In some cases this should speed up access to the information in rpmdb.

6.18.1. Configuration

/etc/yum/pluginconf.d/rpm-warm-cache.conf

```
[main]
enabled = 1
```

6.19. yum-plugin-security

This extension adds **--security**, **--cve**, **--bz** and **--advisory** options to **yum**, plus the commands **list-security** and **info-security**. These options enable the possibility to limit the package listing/updating to the ones relevant for safety reasons. The commands provide the information about security.

6.19.1. Configuration

/etc/yum/pluginconf.d/security.conf

```
[main]
enabled=1
```

enabled = 0|1

Enable extension at a general level.

6.19.2. Usage examples

Security information listing.

```
# yum list-security
Loaded plugins: aliases, protect-packages, protectbase, security, versionlock
1765 packages excluded due to repository protections
Reading version lock configuration

FEDORA-2008-5476 security clamav-0.93.1-1.fc9.i386
FEDORA-2008-5476 security clamav-data-0.93.1-1.fc9.i386
FEDORA-2008-5476 security clamav-filesystem-0.93.1-1.fc9.i386
FEDORA-2008-5476 security clamav-lib-0.93.1-1.fc9.i386
FEDORA-2008-5476 security clamav-server-0.93.1-1.fc9.i386
FEDORA-2008-5476 security clamav-server-sysv-0.93.1-1.fc9.i386
FEDORA-2008-5476 security clamav-update-0.93.1-1.fc9.i386
FEDORA-2008-5072 bugfix cvs-1.11.22-14.fc9.i386
FEDORA-2008-5452 bugfix epiphany-2.22.2-2.fc9.i386
FEDORA-2008-5416 bugfix gnome-python2-2.22.1-2.fc9.i386
FEDORA-2008-5416 bugfix gnome-python2-bonobo-2.22.1-2.fc9.i386
FEDORA-2008-5416 bugfix gnome-python2-canvas-2.22.1-2.fc9.i386
FEDORA-2008-5416 bugfix gnome-python2-gconf-2.22.1-2.fc9.i386
FEDORA-2008-5416 bugfix gnome-python2-gnomevfs-2.22.1-2.fc9.i386
FEDORA-2008-5196 bugfix gtkmm24-2.12.7-1.fc9.i386
FEDORA-2008-4976 bugfix lcms-1.17-5.fc9.i386
FEDORA-2008-4976 bugfix lcms-libs-1.17-5.fc9.i386
FEDORA-2008-5525 bugfix libcurl-7.18.2-1.fc9.i386
FEDORA-2008-5562 bugfix libdrm-2.4.0-0.13.fc9.i386
FEDORA-2008-5448 bugfix libraw1394-1.3.0-7.fc9.i386
FEDORA-2008-5450 bugfix yelp-2.22.1-3.fc9.i386
list-security done

# yum info-security
Loaded plugins: aliases, protect-packages, protectbase, security, versionlock
1765 packages excluded due to repository protections
Reading version lock configuration

=====
clamav-0.93.1-1.fc9
=====
Update ID : FEDORA-2008-5476
Release : Fedora 9
Type : security
Status : stable
Issued : 2008-06-20 16:44:24
Bugs : 451761 - CVE-2008-2713 clamav: DoS / crash via crafted petite file
Description : CVE-2008-2713
Files : clamav-debuginfo-0.93.1-1.fc9.ppc64.rpm
       : clamav-milter-sysv-0.93.1-1.fc9.ppc64.rpm
       : clamav-milter-sendmail-0.93.1-1.fc9.ppc64.rpm
       : clamav-milter-core-0.93.1-1.fc9.ppc64.rpm
...

=====
gnome-python2-2.22.1-2.fc9
=====
Update ID : FEDORA-2008-5416
Release : Fedora 9
Type : bugfix
Status : stable
Issued : 2008-06-20 16:54:54
Bugs : 451754 - /usr/share/gtk-doc should be owned by gnome-python2-gnomevfs
Description : Update to the latest gnome-python 2.22 release.
Files : gnome-python2-debuginfo-2.22.1-2.fc9.ppc64.rpm
       : gnome-python2-devel-2.22.1-2.fc9.ppc64.rpm
       : gnome-python2-gnomevfs-2.22.1-2.fc9.ppc64.rpm
```

```

: gnome-python2-gconf-2.22.1-2.fc9.ppc64.rpm
...
# yum info-sec FEDORA-2008-5676
...
=====
phpMyAdmin-2.11.7-1.fc9
=====
Update ID : FEDORA-2008-5676
Release : Fedora 9
Type : security
Status : stable
Issued : 2008-06-25 00:44:32
Bugs : 452497 - phpMyAdmin < 2.11.7-rc2 contains non-documented security bug
Description : This update solves PMASA-2008-4 (phpMyAdmin security
: announcement) from 2008-06-23: XSS on plausible
: insecure PHP installation; see http://www.phpmyadm
: in.net/home_page/security.php?issue=PMASA-2008-4
: - [interface] New field cannot be auto-increment
: and primary key - [dbi] Incorrect interpretation
: for some mysqli field flags - [display] part 1:
: do not display a TEXT utf8_bin as BLOB (fixed for
: mysqli extension only) - [interface] sanitize the
: after_field parameter, thanks to Norman Hippert -
: [structure] do not remove the BINARY attribute in
: drop-down - [session] Overriding
: session.hash_bits_per_character - [interface]
: sanitize the table comments in table print view,
: thanks to Norman Hippert - [general]
: Auto_Increment selected for TimeStamp by Default -
: [display] No tilde for InnoDB row counter when we
: know it for sure, thanks to Vladyslav Bakayev -
: dandy76 - [display] alt text causes duplicated
: strings - [interface] Cannot upload BLOB into
: existing row - [export] HTML in exports getting
: corrupted, thanks to Jason Judge - jasonjudge -
: [interface] BINARY not treated as BLOB:
: update/delete issues - [general] protection
: against XSS when register_globals is on and
: .htaccess has no effect, thanks to Tim Starling -
: [export] Firefox 3 and .sql.gz (corrupted); detect
: Gecko 1.9, thanks to Juergen Wind
Files : phpMyAdmin-2.11.7-1.fc9.noarch.rpm
: phpMyAdmin-2.11.7-1.fc9.src.rpm
info-sec done

```

Updating only for security reasons.

```

# yum --security update
...
Limiting packages to security relevant ones

Needed 7 (+7 related) of 42 transaction packages, for security
--> Running transaction check
---> Package clamav-server.i386 0:0.93.1-1.fc9 set to be updated
---> Package clamav-lib.i386 0:0.93.1-1.fc9 set to be updated
---> Package clamav-update.i386 0:0.93.1-1.fc9 set to be updated
---> Package clamav-server-sysv.i386 0:0.93.1-1.fc9 set to be updated
---> Package clamav-data.i386 0:0.93.1-1.fc9 set to be updated
---> Package clamav.i386 0:0.93.1-1.fc9 set to be updated
---> Package clamav-filesystem.i386 0:0.93.1-1.fc9 set to be updated
--> Finished Dependency Resolution

Dependencies Resolved

```

```

=====
Package                Arch      Version      Repository    Size
=====
Updating:
clamav                  i386      0.93.1-1.fc9 updates       650 k
clamav-data             i386      0.93.1-1.fc9 updates        14 M
clamav-filesystem       i386      0.93.1-1.fc9 updates        19 k
clamav-lib              i386      0.93.1-1.fc9 updates       301 k
clamav-server           i386      0.93.1-1.fc9 updates        61 k
clamav-server-sysv      i386      0.93.1-1.fc9 updates        18 k
clamav-update           i386      0.93.1-1.fc9 updates        65 k

Transaction Summary
=====
Install      0 Package(s)
Update       7 Package(s)
Remove       0 Package(s)

Total download size: 15 M
Is this ok [y/N]: y

```

6.20. yum-plugin-show-leaves

yum extension that shows leaf packages recently installed and packages that became leaf after a transaction.

6.20.1. Configuration

/etc/yum/pluginconf.d/show-leaves.conf

```

[main]
enabled=1

```

enabled = 0|1

Enable extension at a general level.

6.20.2. Usage examples

Notification at install time.

```

# yum install zsh
...
Dependencies Resolved

=====
Package                Arch      Version      Repository    Size
=====
Instalando:
zsh                    x86_64    4.3.10-4.fc12 fedora        2.0 M

Transaction Summary
=====
Install      1 Package(s)
Update       0 Package(s)

Total download size: 2.0 M
Is this ok [y/N]: y
...

```

```
Warning: Since last transaction, database has been modified (RPMDB).
Instalando      : zsh-4.3.10-4.fc12.x86_64                      1/1

Installed:
zsh.x86_64 0:4.3.10-4.fc12

Complete!
New leaves:
zsh.x86_64          << New leave package, notification
```

6.21. yum-plugin-tmprepo

This extension adds the **--tmprepo** option taking an url of a .repo file, downloads it and enable it for only one transaction. This extension tries to ensure the safe temporary usage of repositories, by default, it does not allow to disable the gpg check.

This extension adds the following options:

--tmprepo=[url]

Enables or creates one or more temporary repository from URLs of the corresponding .repo, repomd.xml, local repomd.xml file or repository local directory.

--tmprepo-keep-created

Keeps the repository configured.

6.21.1. Configuration

/etc/yum/pluginconf.d/tmprepo.conf

```
[main]
enabled=1
# You can uncomment this to allow temporary repositories to disable/alter
# gpg checking. This is not recommended.
# You might want to use pkgs_gpgcheck=false, but if you do understand
# what the security implications are.
```

enabled = 0|1

Enable extension at a general level.

pkgs_gpgcheck=false

Disables the gpg check for packages.

6.21.2. Usage example

With **yum-plugin-tmprepo** enabled and configured with *pkgs_gpgcheck=false* this example shows how to use a remote repository without installing the .repo file:

```
# yum --tmprepo=http://repos.fedorapeople.org/repos/gomix/fox/fedora-fox.repo install adie

Loaded plugins:aliases, presto, refresh-packagekit, tmprepo
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package adie.x86_64 0:3.0.0-1.fc13 set to be installed
--> Processing Dependency: libFOX-1.6.so.0()(64bit) for package: adie-3.0.0-1.fc13.x86_64
--> Running transaction check
---> Package fox.x86_64 0:1.6.38-1.fc13 set to be installed
```

```
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                        Arch      Size      Version
Repository
=====
Installing:
 adie      x86_64    3.0.0-1.fc13    fedora-fox    124 k
Installing for dependencies:
 fox       x86_64    1.6.38-1.fc13   fedora-fox    919 k

Transaction Summary
=====
Install      2 Package(s)

Total download size: 1.0 M
Installed size: 3.8 M
Downloading Packages:
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Ejecutando transacción
  Installing      : fox-1.6.38-1.fc13.x86_64      1/2
  Installing      : adie-3.0.0-1.fc13.x86_64      2/2

Installed:
 adie.x86_64 0:3.0.0-1.fc13

Dependency Installed:
 fox.x86_64 0:1.6.38-1.fc13

Complete!
```

6.22. yum-plugin-tsflags

This extension allows you to specify optional transaction flags (tsflags) on the **yum** command line with the added option **--tsflags**. This option is equivalent to setting *tsflags* in **yum.conf**, but on demand in the command line. These transaction flags are passed to rpm, if you are unsure of what they do, do not use these options.

6.22.1. Configuration

/etc/yum/pluginconf.d/tsflags.conf

```
[main]
enabled=1
```

enabled = 0|1

Enable extension at a general level.

/etc/yum.conf

```
[main]
cachedir=/var/cache/yum/$basearch/$releasever
keepcache=1
gpgcheck=1
plugins=1
installonly_limit=3
```



```
showduplicatesfromrepos=0
tsflags=test
```

Some known flags:

- *noscripts*
- *notriggers*
- *nodocs*
- *test*
- *justdb*
- *repackage*

6.22.2. Usage examples

Install without documentation

```
# yum --tsflags=nodocs install squid
...
Installed:
  squid.x86_64 7:3.1.8-1.fc13
Complete!
```

```
# man squid
No manual entry for squid
```

```
# yum -y reinstall squid
...
Installing:
  squid.x86_64 7:3.1.8-1.fc13
Complete!
```

```
# man squid

squid(8)                                squid(8)

NAME
  squid - proxy caching server
  ...
```

6.23. yum-plugin-upgrade-helper

This extension enables **yum** to remove specific packages when installing/updating based on additional metadata files in the repositories. It is used to simplify the distribution upgrade.

6.23.1. Configuration

`/etc/yum/pluginconf.d/upgrade-helper.conf`

```
[main]
enabled = 1
```

enabled = 0|1

Enable extension at a general level.

6.24. yum-plugin-verify

This extension adds the **verify-multilib**, **verify-all** and **verify-rpm** commands. These work like **rpm -V** to check the package integrity in the system.

yum-plugin-verify also add the options **--verify-filenames** and **--verify-configuration-files**.

yum verify-all [PACKAGE|all|extras]

Check the packages and displays the failed verification data, showing all the differences, even some that would be ignored by rpm.

verify-multilib [PACKAGE|all|extras]

Check packages and shows unsatisfied verification data. **yum verify** is an alias for **yum verify-multilib**.

verify-rpm [PACKAGE|all|extras]

Verifies packages and shows unsatisfied verification data, the output is intended to be similar to **rpm -V**.

--verify-filenames=VERIFY_FILENAMES

It only verifies the specified file names.

--verify-configuration-files=yes|no|true|false

It only checks or not the files labelled as configuration files.

6.24.1. Configuration

/etc/yum/pluginconf.d/verify.conf

```
[main]
enabled=1
# Display verify mismatches with config. files, should probably use AIDE etc.
# instead, if you really want this.
# configuration-files = True

# # Add more colour to your life (these are the defaults):
# [highlight]
#
# # Don't output these problems in colour:
# low-priority = mtime, genchecksum, permissions-missing, state,missingok,ghost
#
# # Don't highlight the values of these problems
# filter-old   = mtime, checksum
# filter-new   = mtime, checksum
#
# # These are the default highlight values
# new         = reverse
#
# old-fg      = red
```

```
#
# file      = underline
# file-fg = green
#
# tags      = bold
# tags-fg  = yellow
# tags-bg  = black
```

6.24.2. Documentation

This extension includes an online man page, see it with the command **man yum-verify**.

6.24.3. Usage examples

The calls without package specification will take some time, please be patient.

```
# yum verify
```

```
# yum verify-all
```

```
# yum verify-rpm
```

An example with a package specified.

```
$ yum verify squid
...
===== Installed Packages =====
squid.x86_64 : The Squid proxy caching server
  File: /usr/share/squid/errors/ar-ae
    Problem: symlink does not match
    Current: ar
    Original:
  File: /usr/share/squid/errors/ar-bh
    Problem: symlink does not match
    Current: ar
    Original:
...
```

An example using the options added by the extension.

```
$ sudo yum verify --verify-filenames='/etc/*' --verify-configuration-files=yes

Loadad plugins:aliases, presto, verify
===== Installed Packages =====
PackageKit-yum-plugin.x86_64 : Tell PackageKit to check for updates when yum
                               : exits
  File: /etc/yum/pluginconf.d/refresh-packagekit.conf
  Tags: configuration
    Problem: checksum does not match
    Current: sha256:f57b424b10afd7ef2d3e14e1fbe7871d51b5ece812d63f223d51210d822cef8f
    Original: sha256:824c5d1b7feb0bf3621bb0b5515d8df6673ae504d9c0fc5fed050070c13d305f
               -----
    Problem: size does not match
    Current: 17
    Original: 18
               -----
    Problem: mtime does not match
```

```
Current: Sun May 16 06:14:24 2010 (108 days, 22:41:18 earlier)
Original: Thu Sep  2 04:55:42 2010

artwiz-aleczapka-anorexia-fonts.noarch : Anorexia font in Artwiz family
File: /etc/fonts/conf.d/60-artwiz-aleczapka-anorexia.conf
Tags: configuration
Problem: symlink does not match
Current: /usr/share/fontconfig/conf.avail/60-artwiz-aleczapka-anorexia.conf
Original:

...

yum-plugin-verify.noarch : Yum plugin to add verify command, and options
File: /etc/yum/pluginconf.d/verify.conf
Tags: configuration
Problem: mtime does not match
Current: Sun Sep 26 07:41:09 2010 (53 days, 19:49:32 later)
Original: Tue Aug  3 11:51:37 2010
verify done
```

6.25. yum-plugin-versionlock

This extension allows that certain specified packages in a file be protected of an eventual update to a newer version.

6.25.1. Configuration

/etc/yum/pluginconf.d/versionlock.conf

```
[main]
enabled = 1
locklist = /etc/yum/pluginconf.d/versionlock.list
# Uncomment this to lock out "upgrade via. obsoletes" etc. (slower)
# follow_obsoletes = 1
```

enabled = 0|1

Enable extension at a general level.

locklist = /path/to/file

It defines the file name with the packages name list (absolute path).

/etc/yum/pluginconf.d/versionlock.list

The file format is one package name per line.

6.25.2. Usage examples

This computer has an available update for **xterm** but it will be blocked.

Before the update blocking.

```
$ yum list updates xterm
xterm.i386                227-1.fc7                updates
```

Current installed version:

```
$ rpm -q xterm
```

```
xterm-225-2.fc7
```

Edit your configuration file to add the file with the package list, add the previous full name, and try to update:

```
$ yum update xterm
Loading "installonlyn" plugin
Loading "versionlock" plugin
Loading "fastestmirror" plugin
Setting up Update Process
Loading mirror speeds from cached hostfile
Reading version lock configuration
Could not find update match for xterm
No Packages marked for Update/Obsoletion
```

It effectively blocks the update for that package, and it does not even list the update for that package.

Only the package

It only blocks the listed package but not its dependants, ie if you wish to update another package which depends on the blocked, and it is necessary to remove the current package to update it, the update of the dependant package will fail. As an example, if **php-5.2.5-7.fc9.i386** is blocked and **yum update** is tried we get the following result.

```
php-5.2.5-7.fc9.i386 from installed has depsolving problems
--> Missing Dependency: php-cli = 5.2.5-7.fc9 is needed by package
php-5.2.5-7.fc9.i386 (installed)
php-5.2.5-7.fc9.i386 from installed has depsolving problems
--> Missing Dependency: php-common = 5.2.5-7.fc9 is needed by package
php-5.2.5-7.fc9.i386 (installed)
Error: Missing Dependency: php-common = 5.2.5-7.fc9 is needed by package
php-5.2.5-7.fc9.i386 (installed)
Error: Missing Dependency: php-cli = 5.2.5-7.fc9 is needed by package
php-5.2.5-7.fc9.i386 (installed)
```

The only known alternative to avoid these errors is to block the dependant packages versions too.

6.26. yum-presto

yum-presto¹ is a **yum** extension that searches for *deltarpm*s instead of *rpms* when these are available in the repositories. This saves a lot of bandwidth when updating.

A *deltarpm* is the difference between two *rpms*. If you have installed **foo-1.0** and **foo-1.1** is available, **yum-presto** will download the *deltarpm* to *foo-1.0 => 1.1* instead of downloading the whole *foo-1.1.rpm*, and will rebuild *foo-1.1* package from the *foo-1.0* installed one and the downloaded *deltarpm*.

6.26.1. Configuration

It is necessary to have repositories with *deltarpm*s enabled. Fedora official repositories offer *drpms* since Fedora 11, so they are configured to use them out of the box.

¹ <https://fedorahosted.org/presto>

/etc/yum/pluginconf.d/presto.conf

```
# Please go to https://fedorahosted.org/presto for a list of presto-enabled
# repositories

[main]
enabled=1
# This defaults to yum's keepcache option, if not set.
# keepdeltas = false

# This lets you change if the delta is downloaded given it's relative size vs.
# the pkg. Eg. the default:
#
# minimum_percentage = 95
#
# ...means that given a pkg of 100M, a delta of 95M (or less) would be
# downloaded instead but a delta of 96M would be skipped in favour of the pkg.
# Percentage of 0 means never use the delta, percentage of 100 means always
# use it (assuming the delta is never bigger than the pkg).
```

enabled = 0|1

Enable extension at a general level.

keepdeltas = true|false

Keep a drpms cache. If no value is set, it uses the **yum** *keepcache* directive setting.

minimum_percentage = #integer

Comparison value against the rpm package to make a decision to choose to download the rpm.
100 means it will always use the deltas, 0 means not to use deltas.

6.26.2. Usage examples

With presto deltarpms:

```
Found deltarpm update for firefox.i386 0:2.0.0.5-1.fc7

Dependencies Resolved

=====
Package                Arch      Version      Repository      Size
=====
Updating:
firefox                i386      2.0.0.5-1.fc7  updates *      1.2 M

Transaction Summary
=====
Install      0 Package(s)
Update      1 Package(s)
Remove      0 Package(s)

Total download size: 1.2 M
Is this ok [y/N]: N
```

Without presto deltarpms:

```
Dependencies Resolved

=====
```

Package	Arch	Version	Repository	Size
=====				
Updating:				
firefox	i386	2.0.0.5-1.fc7	updates	21 M
Transaction Summary				
=====				
Install	0 Package(s)			
Update	1 Package(s)			
Remove	0 Package(s)			
Total download size: 21 M				
Is this ok [y/N]:				

The difference is noteworthy, 21 M vs 1.2 M, which is a big difference. Next the execution.

```
# time yum -y update firefox
Loading "presto" plugin
Setting up Update Process
Loading mirror speeds from cached hostfile
Setting up Presto
Reading Presto metadata in from local files
0 packages excluded due to repository priority protections
0 packages excluded due to repository protections
Resolving Dependencies
--> Running transaction check
---> Package firefox.i386 0:2.0.0.5-1.fc7 set to be updated
Found deltarpm update for firefox.i386 0:2.0.0.5-1.fc7

Dependencies Resolved

=====
Package                Arch      Version      Repository    Size
=====
Updating:
firefox                i386      2.0.0.5-1.fc7 updates *     1.2 M

Transaction Summary
=====
Install      0 Package(s)
Update      1 Package(s)
Remove      0 Package(s)

Total download size: 1.2 M
Downloading Packages:
Downloading DeltaRPMs:
(1/1): firefox-2.0.0.4-2. 100% |=====| 1.2 MB    00:06
Rebuilding rpms from deltarpms
Downloading RPMs:
Running rpm_check_debug
--> Populating transaction set with selected packages. Please wait.
---> Package firefox.i386 0:2.0.0.5-1.fc7 set to be updated
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Updating   : firefox                                     ##### [1/2]
  Cleanup    : firefox                                     ##### [2/2]
Size of all updates downloaded from Presto-enabled repositories: 1.2M
Size of updates that would have been downloaded if Presto wasn't enabled: 21M
This is a savings of 95 percent

Updated: firefox.i386 0:2.0.0.5-1.fc7
Complete!
```

```
real    1m4.198s
user    0m33.223s
sys     0m3.302s
```

Please note that part of the time is used to rebuild the rpm from the deltarpm, and not from bandwidth consumption by downloading, so the time really depends on the computer's processing capabilities.

6.27. PackageKit-yum-plugin

yum-packagekit notifies **PackageKit** to search for updates when **yum** ends. With this, after running **yum update** and installing all the available updates, **gpk-update-icon** will update almost instantly to reflect this change.

6.27.1. Configuration

/etc/yum/pluginconf.d/refresh-packagekit.conf

```
[main]
enabled=1
```

enabled = 0|1

Enable extension at a general level.

6.28. yum-langpacks

langpacks extension for **yum**. **yum-langpacks** is a YUM extension to search for langpacks for your language to be installed.

6.28.1. Configuration

/etc/yum/pluginconf.d/langpacks.conf

```
[main]
enabled=1

# specify locales requiring langpacks
#langpack_locales = ja_JP, zh_CN, mr_IN
langpack_locales = es_ES
```

```
$ yum list available
Loaded plugins:aliases, langpacks, presto, refresh-packagekit
Adding es_ES to language list
...
```

6.29. Yum command usage and its extensions

6.29.1. Disable extensions

In certain circumstances it is useful to temporarily disable the extensions through the command line.

There are two command line options for this purpose:

--disableplugin=plugin

It disables one or more extensions at runtime. The argument is a comma separated list of yum extensions or wildcards.

--noplugins

It disables all extensions at runtime.

6.29.2. Usage examples

```
$ yum --disableplugin=presto list installed
Loaded plugins:aliases, refresh-packagekit
Installed Packages
Abajo.noarch                0.8-6                @/Abajo-0.8-6.noarch
Canna-libs.x86_64          3.7p3-29.fc13        @fedora
ConsoleKit.x86_64          0.4.1-5.fc13         @anaconda-InstallationRepo
ConsoleKit-libs.x86_64     0.4.1-5.fc13         @anaconda-InstallationRepo
ConsoleKit-x11.x86_64      0.4.1-5.fc13         @anaconda-InstallationRepo
...
```

```
$ yum --disableplugin=presto,refresh-* list installed
Loaded plugins:aliases
Installed Packages
Abajo.noarch                0.8-6                @/Abajo-0.8-6.noarch
Canna-libs.x86_64          3.7p3-29.fc13        @fedora
ConsoleKit.x86_64          0.4.1-5.fc13         @anaconda-InstallationRepo
ConsoleKit-libs.x86_64     0.4.1-5.fc13         @anaconda-InstallationRepo
ConsoleKit-x11.x86_64      0.4.1-5.fc13         @anaconda-InstallationRepo
...
```

```
$ yum --noplugins list installed
Installed Packages
Abajo.noarch                0.8-6                @/Abajo-0.8-6.noarch
Canna-libs.x86_64          3.7p3-29.fc13        @fedora
ConsoleKit.x86_64          0.4.1-5.fc13         @anaconda-InstallationRepo
ConsoleKit-libs.x86_64     0.4.1-5.fc13         @anaconda-InstallationRepo
ConsoleKit-x11.x86_64      0.4.1-5.fc13         @anaconda-InstallationRepo
...
```


Yum Utils

There are a number of utilities available in the Fedora package called *yum-utils* and in some other packages. This section is dedicated to the packages and programs provided therein.

Install them with **yum install yum-utils** if the program is provided by that package, if not, use **yum install** with the corresponding program name, package names correspond in all these cases.

7.1. yum-updateonboot

yum-updateonboot is an application provided by a package with the same name that is installed as a service in Fedora and essentially it does **yum update** after boot. This allows machines that have been off for a considerable time to update immediately rather than wait for the **cron** job on early morning.

7.1.1. Configuration

/etc/sysconfig/yum-updateonboot

```
# IF any of these rpms are updated, the yum-updateonboot init script will
# reboot immediately after the yum update. To keep yum-updateonboot from
# rebooting the system, comment this line out.
#REBOOT_RPMS="kernel kernel-smp"

# A list of groups that should be updated at boot. For each group mentioned
# yum-updateonboot will call 'yum -y groupupdate' Since group names tend to
# have spaces in them, used a semi-colon to separate the group names
#GROUPLIST="My Group;MyOtherGroup;Some_Group;My Group 4"

# Be quiet when updating by default
# These options are passed to yum, default value is "-q"
#OPTIONS="-q"
\t\t
```

Like every service in Fedora, it is disabled by default, and it must be configured to start in the desired run levels.

```
$ chkconfig yum-updateonboot on
```

The above command will cause the service to start the next time you boot the system, ie the system will be updated.

7.2. yumdownloader

This utility provided by *yum-utils* package complements the **downloadonly** plugin as it only works if the desired software is not installed. See how it fails and how it can be resolved with **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
```

There is nothing to install, already installed, nothing is downloaded. Try now with **yumdownloader**.

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

Done this, you have the rpm you needed.

Another frequently used option is to specify that you want to download the source rpms:

```
$ yumdownloader --source zsh
...
Enabling fedora-source repository
fedora-source/metalink | 1.7 kB    00:00
zsh-4.3.10-5.fc13.src.rpm | 2.6 MB    00:30
```

Automatic enabling a repository

Note that **yumdownloader** has enabled the "source" repositories to be able to download.

7.3. package-cleanup

This utility provided by *yum-utils* package helps to detect problems in the rpm database and to solve them, below is a listing showing the most common usages.

Orphaned packages

To list the orphaned packages, those packages are not provided by any of the configured repositories, "are repository orphans", use the **--orphans** option.

```
# package-cleanup --orphans
...
8Kingdoms-1.1.0-6.fc9.i386
WorldOfGooDemo-1.41-1.i386
adobe-release-i386-1.0-1.noarch
apollon-1.0.1-14.fc11.i586
...
```

Duplicate Packages

To list the packages with duplicates in the RPM database use the **--dupes** option.

```
$ 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
```

Dependency Problems

To list the dependencies problems in the RPM database use the **--problems** option.

```
$ 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
```

Removing old kernels

package-cleanup is also useful for removing old kernel installations and for removing old kernel related packages use the **--oldkernel** option:

```
# package-cleanup --oldkernel
Setting up yum
Loaded plugins: dellsysidplugin2, fastestmirror, kernel-module, presto,
               : priorities, protectbase, upgrade-helper, versionlock
Loading mirror speeds from cached hostfile
I will remove the following 3 kernel related packages:
kernel-2.6.30.5-43.fc11.i586
kernel-devel-2.6.29.6-217.2.16.fc11.i586
kernel-devel-2.6.30.5-43.fc11.i586
Is this ok [y/N]: y

# rpm -qa kernel*
[root@movix ~]# rpm -qa kernel*
kernel-headers-2.6.30.9-90.fc11.i586
kernel-2.6.30.8-64.fc11.i586
kernel-devel-2.6.30.8-64.fc11.i586
kernel-2.6.30.9-90.fc11.i586
kernel-devel-2.6.30.9-90.fc11.i586
kernel-doc-2.6.30.9-90.fc11.noarch
kernel-firmware-2.6.30.9-90.fc11.noarch
# uname -r
2.6.30.9-90.fc11.i586
```



It does not interpret `installonly_limit`

Note that an old kernel has been left according to the common **yum** practice to maintain two full kernels installed. Beware that this command does not interpret the option `installonly_limit` in the file **yum.conf**.

7.3.1. Usage Examples

```
# 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
...
```

Because of the way the updates are managed, **package-cleanup** only considers the latest version of the package, the outdated packages provided by the configured repositories will appear as orphans, ie the Updates repository for example, does not continue to offer the previous version for the effects of **package-cleanup**. That is, before making a **package-cleanup --orphans**, it is recommended to

make a full update if possible, and to be careful not to remove these packages easily, because they can be manually installed packages working perfectly well, or simply the repository from which they were installed was disabled.

With **package-cleanup** and the **--duplicates** option you can detect duplicate packages in the RPM database, in this example *gedit-2.14.1-1.i386*.

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

We have two installed versions and the older version remained, probably due to some mismanagement. You can now easily start a cleaning job, and that's what **package-cleanup** is about.

```
# 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
```

After this operation is completed, some space on disk has been recovered, has been cleaned up, at least in part, the rpm database that represents the installed software on your Fedora system. For more usage details check the man page, **man package-cleanup**.

7.4. yum-complete-transaction

yum-complete-transaction is provided by *yum-utils* package, and it offers the possibility to finish incomplete yum transactions. This situation is typically necessary, when the yum command displays an output similar to:

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

yum-complete-transaction is a program that searches for incomplete or aborted yum transactions in the system and tries to complete them. Search the **transaction-all*** and **transaction-done*** archives that are normally found in **/var/lib/yum** looking for a transaction that has been aborted in the middle of its execution.

If more than one transaction is not completed, it will attempt to complete the most recent first. You can run this program more than once to clean up all unfinished transactions.

```
$ 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
```

If you do not want to finish any of these transactions you may simply clean the history with **--cleanup-only** option.

```
# yum-complete-transaction --cleanup-only
Loaded plugins: dellsysidplugin2, fastestmirror, kernel-module, presto,
               : priorities, protectbase, upgrade-helper, versionlock
Loading mirror speeds from cached hostfile
* fedora: mirrors.ucr.ac.cr
* livna: rpm.livna.org
* rpmsfusion-free: lordmorgul.net
* rpmsfusion-free-updates: lordmorgul.net
* rpmsfusion-nonfree: lordmorgul.net
* rpmsfusion-nonfree-updates: lordmorgul.net
* updates: mirrors.ucr.ac.cr
0 packages excluded due to repository protections
Reading version lock configuration
Checking for new repos for mirrors
Cleaning up unfinished transaction journals
Cleaning up 2009-09-04.01:09.22
Cleaning up 2009-09-05.07:12.25
```

7.5. yum-builddep

yum-builddep is a program to install the required RPMs to build the specified package. The SRPM package specified should be available in the YUM repository or it may be a local SRPM file. This extension is provided by the *yum-utils* package.

```
# 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]:

```

```

$ sudo yum-builddep --nogpgcheck fox-1.6.33-1.rf.src.rpm
Loaded plugins:presto, refresh-packagekit
Getting requirements for fox-1.6.33-1.rf.src
--> Already installed : gcc-c++-4.4.4-10.fc13.x86_64
--> Already installed : autoconf-2.65-1.fc13.noarch
--> zlib-devel-1.2.3-23.fc12.x86_64
--> bzip2-devel-1.0.6-1.fc13.x86_64
--> 2:libpng-devel-1.2.44-1.fc13.x86_64
--> Already installed : libXext-devel-1.1.2-2.fc13.x86_64
--> Running transaction check
--> Package bzip2-devel.x86_64 0:1.0.6-1.fc13 set to be installed
--> Processing dependencies: bzip2-libs = 1.0.6-1.fc13 for the package: bzip2-
devel-1.0.6-1.fc13.x86_64
--> Package libpng-devel.x86_64 2:1.2.44-1.fc13 set to be installed
--> Package zlib-devel.x86_64 0:1.2.3-23.fc12 set to be installed
--> Running Transaction Test
--> Package bzip2-libs.x86_64 0:1.0.6-1.fc13 set to be updated
--> Finished Dependency Resolution

```

Dependencies Resolved

Package Size	Arch	Version	Repository	
Installing:				
bzip2-devel	x86_64	1.0.6-1.fc13	updates	217 k
libpng-devel	x86_64	2:1.2.44-1.fc1	updates	111 k
zlib-devel	x86_64	1.2.3-23.fc12	fedora	40 k
Updating for dependencies:				
bzip2-libs	x86_64	1.0.6-1.fc13	updates	36 k

Transaction Summary

```

=====
Install      3 Package(s)
Upgrade      1 Package(s)

```

```

Total download size: 405 k
Is this ok [y/N]:

```

What it is left

Note that it only installs the missing packages.

7.6. yum-NetworkManager-dispatcher

This **NetworkManager**'s "dispatch script" forced **yum** to check your cache when establishing a new network connection. Note that currently there is no verification of previous data, so if the WiFi connection passes from top to bottom, or if you suspend / resume often **yum** will check your cache many times.

/etc/NetworkManager/dispatcher.d/yum-NetworkManager-dispatcher is the script for NetworkManager, if you have any other action combined with the cache verification, you could edit this file freely.

7.7. yum-cron

This package contains the files needed to do a **yum update** as a **cron** job. They were introduced in **yum-2.6.1-0.fc5** but have been left in FC6 in favor of **yum-updatesd**. Install this package if you want automatic updates via a nightly **cron** instead of using the latest method with **yum-updatesd** daemon.

Generally **yum-cron** is more suitable for servers and **yum-updatesd** for desktop users, and that's why it is still available.

7.7.1. Start and Stop

The corresponding **cron** job are enabled and disabled like any other daemon:

```
$ sudo yum chkconfig yum-cron on
$ sudo service yum-cron start
```

With the previous couple of commands the file **/etc/cron.daily/0yum.cron** will be daily executed at the scheduled time.

7.7.2. Configuration

/etc/sysconfig/yum-cron

```
# Yum parameters, global at scripts runtime.
YUM_PARAMETER=

# Do no install, just check (yes|no)
CHECK_ONLY=no

# Check if repos are reachable before updating (yes|no)
CHECK_FIRST=no

# Do no install, just check and download (yes|no)
# Implica CHECK_ONLY=yes
DOWNLOAD_ONLY=no

# Verbosity of error reports, range 0-10.
# 0 report only critical errors.
# 1 report all errors.
# ERROR_LEVEL=0

# Debug level, range 0-10.
# Higher value, more verbose debug output.
# Level 0 is the default
# DEBUG_LEVEL=0

# Random wait, 60 implies yum will wait randomly
# between 1 and 60 minutes to act.
RANDOMWAIT="60"

# If defined, MAILTO and mail command is available,
# mail is used to send the output.
# By default, crond sends the output by email by itself
MAILTO=

# SYSTEMNAME , by default hostname
# Only in used in combination with MAILTO
#SYSTEMNAME=""

# Its possible to define weekdays to run the updates
```

```
# by default its every day
#DAYS_OF_WEEK="0123456"

# Cleanup day. By default 0 (Sunday).
# If not in DAYS_OF_WEEK above, it will never clean
CLEANDAY="0"
\t
```

Advanced YUM repository management

We will examine all the native tools available in Fedora for the YUM repository management.

8.1. Repository creation with createrepo

In this section we will create an example yum repository from a set of rpm packages. But first it is necessary to install the **createrepo** tool.

```
# yum install createrepo
```

```
# createrepo -h
Usage: genpkgmetadata.py [options]

Options:
  --version            show program's version number and exit
  -h, --help          show this help message and exit
  -q, --quiet          output nothing except for serious errors
  -v, --verbose        output more debugging info.
  --profile            output timing/profile info.
  -x EXCLUDES, --excludes=EXCLUDES
                        files to exclude
  --basedir=BASEDIR   basedir for path to directories
  -u BASEURL, --baseurl=BASEURL
                        baseurl to append on all files
  -g GROUPFILE, --groupfile=GROUPFILE
                        path to groupfile to include in metadata
  -s SUMTYPE, --checksum=SUMTYPE
                        specify the checksum type to use (default: sha256)
  -p, --pretty         make sure all xml generated is formatted
  -c CACHEDIR, --cachedir=CACHEDIR
                        set path to cache dir
  -C, --checkts       check timestamps on files vs the metadata to see if we
                        need to update
  -d, --database       create sqlite database files
  --update            use the existing repodata to speed up creation of new
  --update-md-path=UPDATE_MD_PATH
                        use the existing repodata for --update from this path
  --skip-stat         skip the stat() call on a --update, assumes if the
                        filename is the same then the file is still the
                        same (only use this if you're fairly trusting or
                        gullible)
  --split             generate split media
  -i PKGLIST, --pkglist=PKGLIST
                        use only the files listed in this file from
                        the directory specified
  -n INCLUDEPKG, --includepkg=INCLUDEPKG
                        add this pkg to the list - can be specified multiple
                        times
  -o OUTPUTDIR, --outputdir=OUTPUTDIR
                        <dir> = optional directory to output to
  -S, --skip-symlinks ignore symlinks of packages
  --changelog-limit=CHANGELOG_LIMIT
                        only import the last N changelog entries
  --unique-md-filenames
                        include the file's checksum in the filename, helps with
                        proxies
  --distro=DISTRO     distro tag and optional cpeid:
                        --distro 'cpeid, textname'
  --content=CONTENT_TAGS
                        tags for the content in the repository
  --revision=REVISION user-specified revision for this repository
```

```
--deltas          create delta rpms and metadata
--oldpackagedirs=OLDPACKAGE_PATHS
                  paths to look for older pkgs to delta against
--num-deltas=NUM_DELTAS
                  the number of older versions to make deltas against
```

The procedure is simple, we have to create a directory to hold the repository, put the rpms provided by developers and packagers there, and then execute **createrepo** specifying the directory where the rpms reside.

```
# mkdir /var/www/html/x86_64/
# cp -ar /home/makerpm/rpmbuild/RPMS/x86_64/*.rpm /var/www/html/x86_64/
# createrepo --verbose /var/www/html/x86_64/
1/14 - dhcp_probe-1.3.0-2.fc13.x86_64.rpm
2/14 - dhcp_probe-debuginfo-1.3.0-2.fc13.x86_64.rpm
3/14 - rubygem-pg-0.9.0-1.fc13.x86_64.rpm
4/14 - dhcp_probe-1.3.0-4.fc13.x86_64.rpm
5/14 - iptables-debuginfo-1.4.5-1.fc12.x86_64.rpm
6/14 - passenger-debuginfo-2.2.15-1.fc13.x86_64.rpm
7/14 - iptables-ipv6-1.4.5-1.fc12.x86_64.rpm
8/14 - dhcp_probe-debuginfo-1.3.0-4.fc13.x86_64.rpm
9/14 - iptables-devel-1.4.5-1.fc12.x86_64.rpm
10/14 - dhcp_probe-debuginfo-1.3.0-3.fc13.x86_64.rpm
11/14 - dhcp_probe-1.3.0-3.fc13.x86_64.rpm
12/14 - dhcp_probe-debuginfo-1.3.0-1.fc13.x86_64.rpm
13/14 - passenger-2.2.15-1.fc13.x86_64.rpm
14/14 - iptables-1.4.5-1.fc12.x86_64.rpm

Saving Primary metadata
Saving file lists metadata
Saving other metadata
```

It can be easily checked that the previous process has created a **repodata** directory with the metadata representing and conforming the new repository.

```
# ls /var/www/html/x86_64/repodata/
filelists.xml.gz  other.xml.gz  primary.xml.gz  repomd.xml
```

Once the metadata for the YUM repository is created, it is time to make it available. In this example we will create a local configuration using the `file:///` access method that will allow us to access the recently created repository.

/etc/yum.repos.d/rubyera.repo

```
[rubyera]
name=Rubyera repo
baseurl=file:///var/www/html/x86_64
enabled=1
gpgcheck=0
```

To verify the availability of the new repository the **repoquery** tool can be used.

```
# repoquery -q --repoid=rubyera -a
dhcp_probe-0:1.3.0-4.fc13.x86_64
dhcp_probe-debuginfo-0:1.3.0-4.fc13.x86_64
iptables-0:1.4.5-1.fc12.x86_64
```

```
iptables-debuginfo-0:1.4.5-1.fc12.x86_64
iptables-devel-0:1.4.5-1.fc12.x86_64
iptables-ipv6-0:1.4.5-1.fc12.x86_64
passenger-0:2.2.15-1.fc13.x86_64
passenger-debuginfo-0:2.2.15-1.fc13.x86_64
rubygem-pg-0:0.9.0-1.fc13.x86_64
```

8.2. Query repositories with repoquery

repoquery is a program to query information from YUM repositories, in a similar way as the queries with **rpm**.

repoquery is a YUM tool with a set of deep and complicated options. It is intended to be similar to **rpm -q** but over remote repositories. This explains its complexity. Some pero ejecutada sobre los repositorios remotos. Esto explica el por qué de lo complicado y profundo. Below are some common uses.

By default **repoquery** acts over configured and enabled repositories in **yum** configuration. Repositories can be enabled/disabled with the **--enablerepo=repoid** and **--disablerepo=repoid** command line options. Occasionally it is only needed to query a particular repository, for this situation the option **--repoid=repoid** must be used. With this last option it will be queried only that repository regardless of **yum** settings.

Sometimes you will need to query an unconfigured repository. **repoquery** handles that situation too with the use of **--repofrompath** option:

```
$ repoquery --repofrompath=reponame,http://path/to/the/repo --repoid=reponame -q -a
```

repoquery can work over any architecture, and uses the system's architecture where it is run from by default. However it is possible to specify other architectures with **--archlist** command line option.

Just i686 packages:

```
$ repoquery -q --archlist=i686 -a
```

Just i686 and i386 packages:

```
$ repoquery -q --archlist=i686,i386 -a
```

Only srpms source packages:

```
$ repoquery -q --archlist=src -a
```

List the contents of a given package.

```
$ repoquery --list iftop
/usr/sbin/iftop
/usr/share/doc/iftop-0.17
/usr/share/doc/iftop-0.17/COPYING
/usr/share/doc/iftop-0.17/ChangeLog
/usr/share/doc/iftop-0.17/README
/usr/share/doc/iftop-0.17/TODO
/usr/share/man/man8/iftop.8.gz
```

Chapter 8. Advanced YUM repository management

List the packages owner of a given installed file.

```
$ repoquery -q --file /usr/bin/yum
yum-0:3.2.25-1.fc12.noarch
yum-0:3.2.27-3.fc12.noarch
```

Use of **repoquery** limiting it to installed packages.

```
$ repoquery --pkgnarrow=installed -q --whatrequires --alldeps lsof
rkhunter-0:1.3.6-7.fc12.noarch
rkhunter-0:1.3.4-8.fc12.noarch
libguestfs-1:1.0.75-1.fc12.i686
libguestfs-1:1.2.7-1.fc12.1.i686
```

If you need to find out which srpms have a build requirement on a particular package, run:

```
$ repoquery --archlist=src --repoid=fedora-source -q --whatrequires lsof
libguestfs-1:1.0.75-1.fc12.src
```

If you need to find out the dependence tree of a given package, run the command:

```
$ repoquery --tree-requires fwsnort

fwsnort-1.0.6-7.fc13.noarch [cmd line]
├─ bash-4.1.7-1.fc13.x86_64 [1: /bin/sh]
│ └─ glibc-2.12.1-2.i686 [1: rtld(GNU_HASH)]
│   └─ basesystem-10.0-3.noarch [1: basesystem]
│     └─ filesystem-2.4.31-1.fc13.x86_64 [1: filesystem]
│       └─ setup-2.8.20-1.fc13.noarch [1: setup >= 2.5.4-1]
│         └─ setup-2.8.20-1.fc13.noarch [1: setup]
│       ...
│     ...
│   └─ 4:perl-libs-5.10.1-119.fc13.x86_64 [3: libperl.so()(64bit), perl-libs, perl-libs =
4:5.10.1-119.fc13]
│     └─ 3:perl-version-0.82-1.fc13.x86_64 [1: perl(version)]
│       └─ perl-IPTables-Parse-0.7-5.fc13.noarch [1: perl-IPTables-Parse]
│         └─ 4:perl-5.10.1-112.fc13.i686 [7: perl >= 5.006, perl(:MODULE_COMPAT_5.10.1),
perl(Carp), perl(POSIX), perl(strict), perl(vars), perl(warnings)]
│           └─ 4:perl-5.10.1-119.fc13.x86_64 [7: perl >= 5.006, perl(:MODULE_COMPAT_5.10.1),
perl(Carp), perl(POSIX), perl(strict), perl(vars), perl(warnings)]
│             └─ perl-Net-IPv4Addr-0.10-7.fc13.noarch [1: perl-Net-IPv4Addr]
│               └─ 4:perl-5.10.1-112.fc13.i686 [6: /usr/bin/perl, perl(:MODULE_COMPAT_5.10.1),
perl(Carp), perl(Getopt::Long), perl(strict), perl(vars)]
│                 └─ 4:perl-5.10.1-119.fc13.x86_64 [6: /usr/bin/perl, perl(:MODULE_COMPAT_5.10.1),
perl(Carp), perl(Getopt::Long), perl(strict), perl(vars)]
```

If you need a formatted package list, with their url and the repository where they came from, you must **repoquery** as follows:

```
$ repoquery -q --qf="%{name} %{repoid} %{location}" pkg1 pkg2 pkg3
```

Query the packages with a pattern to a specific repository:

```
$ repoquery -q --repoid=updates-testing fw*
fwsnort-0:1.0.6-7.fc13.noarch
```

For more details please see the man page, `man repoquery`.

8.3. Groups management with yum-groups-manager

yum-groups-manager is used to create or edit the groups metadata file of a YUM repository. Using this tool is often much easier than to edit the corresponding XML by hand. **yum-groups-manager** can load a whole group metadata file, create a new group or edit an existing one, and then write the metadata as output.

Before you can use this tool you will need that the packages to be added to the corresponding groups be available via **yum**, ie, they must be available in the configured repositories.

Creation of the Rubyera YUM group

```
# yum-groups-manager -n "Rubyera" --id=rubyera --description="Rubyera, packages from..." --
save=comps.xml --mandatory passenger
```

```
# cat comps.xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE comps PUBLIC "-//Red Hat, Inc.//DTD Comps info//EN" "comps.dtd">
<comps>

  <group>
    <id>rubyera</id>
    <default>false</default>
    <uservisible>true</uservisible>
    <display_order>1024</display_order>
    <name>Rubyera</name>
    <description>Rubyera, packages from...</description>
    <packagelist>
      <packagereq type="mandatory">passenger</packagereq>
    </packagelist>
  </group>
</comps>
```

Once created the **comps.xml** file and after placing it in the same directory with the rpms, you must run the **createrepo** command now specifying the name of the file that defines groups.

```
# createrepo --verbose -g comps.xml /var/www/html/x86_64/
```

Now the group is available.

```
# yum clean metadata
# yum groupinfo Rubyera
Loaded plugins: download-order, downloadonly, filter-data, list-data, presto, verify
Setting up Group Process
...
rubyera
| 1.9 kB    00:00 ...
rubyera/group_gz
| 341 B    00:00 ...

Group: Rubyera
Description: Rubyera, paquetes producidos por ...
Mandatory Packages:
  passenger
```

One application type at a time

It is not possible to add mandatory applications and optionals or by default in one command, you must use separate commands, for example, to add optional packages you can do:

```
# yum-groups-manager --id=rubyera --load=repodata/comps.xml --save=comps.xml --optional
dhcp_probe
```

Note that you can not operate on the same input and output file. The option **--load** loads the **comps.xml** file previously produced by **createrepo**, while **--save** is operating on the directory where the rpms are, in this case **/var/www/html/x86_64**.

Once you have a new **comps.xml** file, shown below, you must rebuild the YUM repository with **createrepo**.

```
# cat rubyera.xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE comps PUBLIC "-//Red Hat, Inc.//DTD Comps info//EN" "comps.dtd">
<comps>

  <group>
    <id>rubyera</id>
    <default>false</default>
    <uservisible>true</uservisible>
    <display_order>1024</display_order>
    <name>Rubyera</name>
    <description>Rubyera, paquetes producidos por...</description>
    <packagelist>
      <packagereq type="optional">dhcp_probe</packagereq>
      <packagereq type="mandatory">passenger</packagereq>
    </packagelist>
  </group>
</comps>
```

```
# createrepo --verbose -g comps.xml /var/www/html/x86_64/
```

Verification of the new composition of Rubyera group.

```
# yum clean metadata
# yum groupinfo Rubyera
...
Group: Rubyera
Description: Rubyera, paquetes producidos por ...
Mandatory Packages:
  passenger
Optional Packages:
  dhcp_probe
```

To add a second group:

```
# yum-groups-manager --name="Rubyera debuginfo" --id=rubyera-debuginfo --load=repodata/
comps.xml --save=comps.xml dhcp_probe-debuginfo passenger-debuginfo
```



```
# cat comps.xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE comps PUBLIC "-//Red Hat, Inc.//DTD Comps info//EN" "comps.dtd">
<comps>

  <group>
    <id>rubyera</id>
    <default>>false</default>
    <uservisible>>true</uservisible>
    <display_order>1024</display_order>
    <name>Rubyera</name>
    <description>Rubyera, packages from...</description>
    <packagelist>
      <packagereq type="mandatory">passenger</packagereq>
      <packagereq type="optional">dhcp_probe</packagereq>
    </packagelist>
  </group>
  <group>
    <id>rubyera-debuginfo</id>
    <default>>false</default>
    <uservisible>>true</uservisible>
    <display_order>1024</display_order>
    <name>Rubyera debuginfo</name>
    <description></description>
    <packagelist>
      <packagereq type="default">dhcp_probe-debuginfo</packagereq>
      <packagereq type="default">passenger-debuginfo</packagereq>
    </packagelist>
  </group>
</comps>
```

```
# yum groupinfo "Rubyera debuginfo"
...
Group: Rubyera debuginfo
Default Packages:
  dhcp_probe-debuginfo
  passenger-debuginfo
```

8.4. Creating YUM mirrors with reposync

reposync is used to synchronize a remote YUM repository with a local directory using **yum** to get the packages.

Usage:

```
reposync [options]
```

Options:

```
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
```

It is trivial to reconstruct the metadata with **createrepo** from a local directory with the rpms.

Example usage to synchronize the rubyera repository.

```
$ reposync --repoid=rubyera
rubyera                               | 3.2 kB      00:00 ...
[rubyera: 1      of 14    ] Downloading dhcp_probe-1.3.0-2.fc13.x86_64.rpm
[rubyera: 2      of 14    ] Downloading dhcp_probe-1.3.0-4.fc13.x86_64.rpm
[rubyera: 3      of 14    ] Downloading dhcp_probe-1.3.0-3.fc13.x86_64.rpm
[rubyera: 4      of 14    ] Downloading dhcp_probe-debuginfo-1.3.0-2.fc13.x86_64.rpm
[rubyera: 5      of 14    ] Downloading dhcp_probe-debuginfo-1.3.0-4.fc13.x86_64.rpm
[rubyera: 6      of 14    ] Downloading dhcp_probe-debuginfo-1.3.0-3.fc13.x86_64.rpm
[rubyera: 7      of 14    ] Downloading dhcp_probe-debuginfo-1.3.0-1.fc13.x86_64.rpm
[rubyera: 8      of 14    ] Downloading iptables-1.4.5-1.fc12.x86_64.rpm
[rubyera: 9      of 14    ] Downloading iptables-debuginfo-1.4.5-1.fc12.x86_64.rpm
[rubyera: 10     of 14    ] Downloading iptables-devel-1.4.5-1.fc12.x86_64.rpm
[rubyera: 11     of 14    ] Downloading iptables-ipv6-1.4.5-1.fc12.x86_64.rpm
[rubyera: 12     of 14    ] Downloading passenger-2.2.15-1.fc13.x86_64.rpm
[rubyera: 13     of 14    ] Downloading passenger-debuginfo-2.2.15-1.fc13.x86_64.rpm
[rubyera: 14     of 14    ] Downloading rubygem-pg-0.9.0-1.fc13.x86_64.rpm
```

If you try to synchronize again, **reposync** will not download the existing packages saving bandwidth.

```
$ reposync --repoid=rubyera
[rubyera: 1      of 14    ] Skipping existing dhcp_probe-1.3.0-2.fc13.x86_64.rpm
[rubyera: 2      of 14    ] Skipping existing dhcp_probe-1.3.0-4.fc13.x86_64.rpm
[rubyera: 3      of 14    ] Skipping existing dhcp_probe-1.3.0-3.fc13.x86_64.rpm
[rubyera: 4      of 14    ] Skipping existing dhcp_probe-debuginfo-1.3.0-2.fc13.x86_64.rpm
[rubyera: 5      of 14    ] Skipping existing dhcp_probe-debuginfo-1.3.0-4.fc13.x86_64.rpm
[rubyera: 6      of 14    ] Skipping existing dhcp_probe-debuginfo-1.3.0-3.fc13.x86_64.rpm
[rubyera: 7      of 14    ] Skipping existing dhcp_probe-debuginfo-1.3.0-1.fc13.x86_64.rpm
[rubyera: 8      of 14    ] Skipping existing iptables-1.4.5-1.fc12.x86_64.rpm
[rubyera: 9      of 14    ] Skipping existing iptables-debuginfo-1.4.5-1.fc12.x86_64.rpm
[rubyera: 10     of 14    ] Skipping existing iptables-devel-1.4.5-1.fc12.x86_64.rpm
[rubyera: 11     of 14    ] Skipping existing iptables-ipv6-1.4.5-1.fc12.x86_64.rpm
[rubyera: 12     of 14    ] Skipping existing passenger-2.2.15-1.fc13.x86_64.rpm
[rubyera: 13     of 14    ] Skipping existing passenger-debuginfo-2.2.15-1.fc13.x86_64.rpm
[rubyera: 14     of 14    ] Skipping existing rubygem-pg-0.9.0-1.fc13.x86_64.rpm
```

An example use is to create a local mirror for the fedora and fedora-updates repositories and then locally. It would be extremely quick to install and upgrade packages in this way. The important thing is to maintain regularly synchronized the local mirror with the master repository running every hour **reposync** and **createrepo**. It is important to note that the groups file is not downloaded by **reposync**. Use **wget** or **curl** to download it.

One of the advantages of using **reposync** instead of such mirrored with rsync, ftp or http, is that **reposync** benefits from the use of the list of mirrors in the definition of the repository you want to

mirror by using **yum**. In return, when you use rsync, ftp or http, it is dependent on a single master server, and if it fails, it is more difficult to recover from such failure and handle it, while with **reposync** is totally transparent and simple to handle failures.

8.5. Web browsable repositories with repoview

repoview makes the YUM repository browsable with a number of static HTML pages. It uses the kid templating engine to create the pages so it's customizable. To install it:

```
# yum install repoview
```

repoview requires the repository to have the metadata in sqlite file, if it has not, you will get an error like the following.

```
# repoview /var/www/html/x86_64
Examining repository...Sorry, sqlite files not found in the repository.
Please rerun createrepo with a -d flag and try again.
```

Correct and rebuild the YUM repository using **--database** to generate the sqlite files.

```
# createrepo --verbose --database -g comps.xml /var/www/html/x86_64
1/14 - dhcp_probe-1.3.0-2.fc13.x86_64.rpm
2/14 - dhcp_probe-debuginfo-1.3.0-2.fc13.x86_64.rpm
3/14 - rubygem-pg-0.9.0-1.fc13.x86_64.rpm
4/14 - dhcp_probe-1.3.0-4.fc13.x86_64.rpm
5/14 - iptables-debuginfo-1.4.5-1.fc12.x86_64.rpm
6/14 - passenger-debuginfo-2.2.15-1.fc13.x86_64.rpm
7/14 - iptables-ipv6-1.4.5-1.fc12.x86_64.rpm
8/14 - dhcp_probe-debuginfo-1.3.0-4.fc13.x86_64.rpm
9/14 - iptables-devel-1.4.5-1.fc12.x86_64.rpm
10/14 - dhcp_probe-debuginfo-1.3.0-3.fc13.x86_64.rpm
11/14 - dhcp_probe-1.3.0-3.fc13.x86_64.rpm
12/14 - dhcp_probe-debuginfo-1.3.0-1.fc13.x86_64.rpm
13/14 - passenger-2.2.15-1.fc13.x86_64.rpm
14/14 - iptables-1.4.5-1.fc12.x86_64.rpm

Saving Primary metadata
Saving file lists metadata
Saving other metadata
Generating sqlite DBs
Starting other db creation: Sun Aug 15 07:52:45 2010
Ending other db creation: Sun Aug 15 07:52:46 2010
Starting filelists db creation: Sun Aug 15 07:52:46 2010
Ending filelists db creation: Sun Aug 15 07:52:46 2010
Starting primary db creation: Sun Aug 15 07:52:46 2010
Ending primary db creation: Sun Aug 15 07:52:48 2010
Sqlite DBs complete
```

Now it is possible to run **repoview** successfully, below is a simple example.

```
# repoview --title "rubyera repo for F13 x86_64" /var/www/html/x86_64
Examining repository...done
Opening primary database...done
Opening changelogs database...done
Parsing comps.xml...done
Examining state db...done
Collecting letters...done
Writing package dhcp_probe.html
```

```
Writing package passenger.html
Writing group rubyera.group.html
Writing package dhcp_probe-debuginfo.html
Writing package passenger-debuginfo.html
Writing group rubyera-debuginfo.group.html
Writing group letter_d.group.html
Writing package iptables.html
Writing package iptables-debuginfo.html
Writing package iptables-devel.html
Writing package iptables-ipv6.html
Writing group letter_i.group.html
Writing group letter_p.group.html
Writing package rubygem-pg.html
Writing group letter_r.group.html
Collecting latest packages...done
Writing index.html...done
```

```
# ls
comps.xml
dhcp_probe-1.3.0-2.fc13.x86_64.rpm
...
passenger-2.2.15-1.fc13.x86_64.rpm
repodata
repoview                                << new directory
rubygem-pg-0.9.0-1.fc13.x86_64.rpm
```

Pointing a web browser to the `/var/www/html/x86_64/repoview/index.html` file you will get a visual result like the following.



8.6. Unresolved dependencies with repoclosure

repoclosure reads the metadata from a number of repositories and checks all the dependencies in all packages and their resolution. Print a list of packages with unresolved dependencies.

Usage:

```
$ repoclosure -h
Usage:
    Read in the metadata of a series of repositories and check all the
    dependencies in all packages for resolution. Print out the list of
```

```

packages with unresolved dependencies

/usr/bin/repoclosure [-c <config file>] [-a <arch>] [-r <repoid>] [-r <repoid2>]

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      check packages of the given archs, can be specified
                           multiple times (default: current arch)
--basearch=BASEARCH       set the basearch for yum to run as
-b, --builddeps           check build dependencies only (needs source repos
                           enabled)
-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
-q, --quiet               quiet (no output to stderr)
-n, --newest              check only the newest packages in the repos
--repofrompath=REPOFROMPATH
                           specify repoid & paths of additional repositories -
                           unique repoid and path required, can be specified
                           multiple times. Example.
                           --repofrompath=myrepo,/path/to/repo
-p PKG, --pkg=PKG         check closure for this package only
-g GROUP, --group=GROUP   check closure for packages in this group only

```

Usage example:

```

# repoclosure -r rubyera
Reading in repository metadata - please wait...
Checking Dependencies
Repos looked at: 1
  rubyera
Num Packages in Repos: 14
package: dhcp_probe-1.3.0-2.fc13.x86_64 from rubyera
  unresolved deps:
    libc.so.6(GLIBC_2.2.5)(64bit)
    libc.so.6()(64bit)
    rtld(GNU_HASH)
    libc.so.6(GLIBC_2.3.4)(64bit)
    libc.so.6(GLIBC_2.7)(64bit)
    libc.so.6(GLIBC_2.4)(64bit)
    libpcap.so.1()(64bit)
    libnet.so.1()(64bit)
package: dhcp_probe-1.3.0-3.fc13.x86_64 from rubyera
  unresolved deps:
    libc.so.6(GLIBC_2.2.5)(64bit)
    libc.so.6()(64bit)
    rtld(GNU_HASH)
    libc.so.6(GLIBC_2.3.4)(64bit)
    libc.so.6(GLIBC_2.7)(64bit)
    libc.so.6(GLIBC_2.4)(64bit)
    libpcap.so.1()(64bit)
    libnet.so.1()(64bit)

```

8.7. Package tracking with repotrack

repotrack keeps aware of any package and its dependencies. It will download the packages you want to monitor and all its dependencies recursively.

Usage:

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

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  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
```

Before you start a download, try running it with **-u** option to get a list of urls to be downloaded. You might have some surprises as **repotrack** will consider to download all provider for a given dependency.

```
$ repotrack -u fwsnort
http://download.fedoraproject.org/pub/fedora/linux/releases/13/Everything/x86_64/os/Packages/
MAKEDEV-3.24-4.x86_64.rpm
http://download.fedoraproject.org/pub/fedora/linux/updates/13/x86_64/
amanda-2.6.1p2-4.fc13.x86_64.rpm
http://download.fedoraproject.org/pub/fedora/linux/updates/13/x86_64/
amanda-2.6.1p2-4.fc13.i686.rpm
...
http://download.fedoraproject.org/pub/fedora/linux/updates/13/x86_64/
xinetd-2.3.14-32.fc13.x86_64.rpm
http://download.fedoraproject.org/pub/fedora/linux/releases/13/Everything/x86_64/os/Packages/
zlib-1.2.3-23.fc12.x86_64.rpm
http://download.fedoraproject.org/pub/fedora/linux/releases/13/Everything/x86_64/os/Packages/
zlib-1.2.3-23.fc12.i686.rpm
```

PackageKit CLI

This manual has deepened in the YUM system as a native tool for software management in the command line, but it is interesting to note that **PackageKit CLI** suite offers the promise of universal commands at command line across multiple Linux distributions. **PackageKit CLI** has relatively little documentation and it is not the main focus of PackageKit and project development, so we will give an overview of their current capabilities.

9.1. pkcon

pkcon is the PackageKit's command line interface or software management CLI.

In general, actions that require privileges will require you to provide the root password at the graphical environment, something not so orthodox as we are using a CLI tool. You can also use **sudo** on the command line to run **pkcon** to gain the privileges or simply run the command directly as root with the help of **su** or after a root shell login via a virtual console or other remote console mechanism.

Basically examples

Because there is no further documentation, the most of the text below are examples of use for common operations. Also note that some subcommands simply do not work.

9.1.1. Documentation

Online Help

```
Usage:
  pkcon [OPTION...] PackageKit Console Program

PackageKit Console Interface

Subcommands:
  get-actions
  get-groups
  get-filters
  get-transactions
  get-time
  search [name|details|group|file] [data]
  install [packages]
  install-local [files]
  download [directory] [packages]
  install-sig [type] [key_id] [package_id]
  remove [package]
  update <package>
  refresh [--force]
  resolve [package]
  get-updates
  get-depends [package]
  get-requires [package]
  get-details [package]
  get-distro-upgrades
  get-files [package]
  get-update-detail [package]
  get-packages
  repo-list
  repo-enable [repo_id]
  repo-disable [repo_id]
  repo-set-data [repo_id] [parameter] [value];
```

```
what-provides [search]
get-categories
```

Help Options:

-h, --help	Show help options
--help-all	Show all help options
--help-debug	Show debugging options

Application Options:

--version	Show the program version and exit
--filter	Set the filter, e.g. installed
--root	Set the install root, e.g. '/' or '/mnt/ltsp'
-n, --nowait	Exit without waiting for actions to complete
-y, --noninteractive	Install the packages without asking for confirmation
--background	Run the command using idle network bandwidth and also using less power
-p, --plain	Print to screen a machine readable output, rather than using animated widgets
-v, --verbose	Show debugging information for all files

9.1.2. Searches

Search by Name

Search by pattern matching in the package name.

search [*name*|*details*|*group*|*file*] [*data*]

```
$ pkcon search name octa

Transaction:\tSearching by name
Status: \tStarting
Status: \tQuerying
Package:\tGMT-octave-4.5.3-3.fc14.x86_64
Package:\toctave-6:3.2.4-3.fc14.i686
Package:\toctave-6:3.2.4-3.fc14.x86_64
...
Package:\tpfstools-octave-1.8.1-1.fc14.2.x86_64
Package:\tplplot-octave-5.9.6-5.fc14.x86_64
Package:\tqtoctave-0.9.1-2.fc14.x86_64
...
Available    pfstools-octave-1.8.1-1.fc14.2.x86_64
Available    plplot-octave-5.9.6-5.fc14.x86_64
Available    qtoctave-0.9.1-2.fc14.x86_64
```

9.1.3. Install, update and remove packages

Install packages from the repositories

```
$ su -c "pkcon install qtoctave"

Password:
Transaction:\tInstalling
Status: \tWaiting for authentication
Percentage:\t0
Status: \tRunning
Status: \tResolving dependencies
Percentage:\t10
Status: \tDownloading packages
Percentage:\t40
```



```

Status: \tChecking signatures
Status: \tTesting changes
Package:\tsuitesparse-3.4.0-2.fc12.x86_64
Status: \tInstalling packages
Percentage:\t70
Package:\toctave-6:3.2.4-3.fc14.x86_64
Percentage:\t80
Package:\tqtoctave-0.9.1-2.fc14.x86_64
Percentage:\t100
Status: \tScanning applications
Status: \tGetting information
Results:
Installed      octave-6:3.2.4-3.fc14.x86_64
Installed      qtoctave-0.9.1-2.fc14.x86_64
Installed      suitesparse-3.4.0-2.fc12.x86_64

```

Install packages from rpm files

```

$ su -c "pkcon install-local /home/makerpm/rpmbuild/RPMS/x86_64/fox-1.6.38-1.fc13.x86_64.rpm"
Password:
Transaction:\tInstalling files
Status: \tWaiting for authentication
Status: \tStarting
Percentage:\t0
Status: \tRunning
Transaction:\tInstalling files
Status: \tWaiting for authentication
Percentage:\t0
Status: \tRunning
Status: \tResolving dependencies
Status: \tTesting changes
Package:\tfox-1.6.38-1.fc13.x86_64
Status: \tInstalling packages
Percentage:\t100
Results:
Installed      fox-1.6.38-1.fc13.x86_64

```

Update packages

```

$ su -c "pkcon update binutils"
Password:
Transaction:\tUpdating packages
Status: \tWaiting for authentication
Status: \tChecking applications in use
Percentage:\t0
Status: \tRunning
Status: \tResolving dependencies
Percentage:\t10
Status: \tDownloading packages
Status: \tTesting changes
Package:\tbinutils-2.20.51.0.7-6.fc14.x86_64
Status: \tInstalling updates
Percentage:\t60
Percentage:\t70
Package:\tbinutils
Status: \tCleaning up packages
Status: \tChecking applications in use
Status: \tGetting information
Results:
Updated        binutils-2.20.51.0.7-6.fc14.x86_64

```

Remove packages

```
$ su -c "pkcon remove suitesparse"

Password:
Transaction:\tRemoving
Status: \tWaiting for authentication
Status: \tStarting
Percentage:\t0
Status: \tRunning
Status: \tResolving dependencies
Status: \tTesting changes
Package:\tqt octave
Percentage:\t70
Status: \tRemoving packages
Package:\toctave
Package:\tsuitesparse
Percentage:\t100
Results:
Removed      octave
Removed      qt octave
Removed      suitesparse
```

9.1.4. Packages Info

Listing all packages

```
$ pkcon get-packages
Transaction:  Getting packages
Status:      Starting
Status:      Querying
Package:     libconfig-1.4.5-1.fc14.x86_64
Package:     clucene-core-0.9.21b-2.fc14.x86_64
Package:     ncurses-5.7-8.20100703.fc14.x86_64
Package:     fontconfig-2.8.0-2.fc14.x86_64
Package:     telepathy-filetransfer-0.0.2-1.fc12.noarch
Package:     python-libs-2.7-8.fc14.1.x86_64
...
Available    Frodo-gui-4.1b-6.fc11.x86_64
Installed    GConf2-2.31.91-1.fc14.x86_64
Installed    GConf2-2.31.91-1.fc14.i686
Available    GConf2-devel-2.31.91-1.fc14.x86_64
Available    GConf2-devel-2.31.91-1.fc14.i686
Installed    GConf2-gtk-2.31.91-1.fc14.x86_64
Available    GLC_Player-2.0.0-2.fc12.x86_64
...
Installed    zziplib-0.13.49-8.fc12.x86_64
Available    zziplib-0.13.49-8.fc12.i686
Available    zziplib-devel-0.13.49-8.fc12.i686
Available    zziplib-devel-0.13.49-8.fc12.x86_64
Available    zziplib-utils-0.13.49-8.fc12.x86_64
Available    zzuf-0.13-1.fc13.x86_64
```

Package Details

```
$ pkcon get-details fwsnort
```

```

Transaction:\tGetting details
Status: \tStarting
Status: \tGetting information
Results:
Package description
  package:    fwsnort-1.0.6-8.fc14.noarch
  license:    GPLv2+
  group:      other
  description: fwsnort translates Snort rules into equivalent iptables rules and generates
a Bourne shell script that implements the resulting iptables commands.

In addition, fwsnort (optionally) uses the IPTables::Parse module to parse the
iptables ruleset on the machine to determine which Snort rules are applicable
to the specific iptables policy.

fwsnort is able to translate approximately 60% of all rules from the
Snort-2.3.3 IDS into equivalent iptables rules.
  size:      5246625 bytes
  url:       http://www.cipherdyne.org/fwsnort/

```

Update details of a package

It is not necessary to have the package previously installed to see this information.

```

$ pkcon get-update-detail fwsnort

Transaction:\tGetting update details
Status: \tStarting
Status: \tGetting information
Results:
Details about the update:
  Package: fwsnort-1.0.6-8.fc14.noarch
  Updates: fwsnort;1.0.6-8.fc14;noarch;installed
  Obsoletes:
  Vendor: https://admin.fedoraproject.org/updates/F14/FEDORA-2010-16198;Fedora 14 Update
FEDORA-2010-16198
  Bugzilla:
  CVE:
  Update text: Minor spec adjust for improved perl packaging guidelines compliance.
  Changes:
  State: stable
  Issued: 2010-10-13T06:08:15
  Updated:

```

9.1.5. General Update

```

$ su -c "pkcon update"
Password:
Transaction: Updating system
Status: Starting
Status: Downloading repository information
Status: Downloading list of packages
Percentage: 0
Status: Running
Status: Resolving dependencies
Percentage: 10
Status: Downloading packages
Percentage: 40

```

```
...
Package: ncurses-libs-5.7-9.20100703.fc14.i686
Package: selinux-policy-targeted-3.9.7-18.fc14.noarch
Package: git-gui-1.7.3.4-1.fc14.noarch
Package: cronie
Status: Cleaning up packages
Package: cronie-anacron
Package: ncurses
Package: ncurses-libs
Package: ImageMagick-perl
Package: ImageMagick-c++
Package: ncurses-libs
Percentage: 80
...\
Updated      cronie-anacron-1.4.5-4.fc14.x86_64
Cleaned up   cronie-anacron
Updated      git-1.7.3.4-1.fc14.x86_64
Cleaned up   git
Updated      git-gui-1.7.3.4-1.fc14.noarch
Cleaned up   git-gui
Updated      gitk-1.7.3.4-1.fc14.noarch
Cleaned up   gitk
...\
Updated      perl-Git-1.7.3.4-1.fc14.noarch
Cleaned up   perl-Git
Updated      selinux-policy-3.9.7-18.fc14.noarch
Cleaned up   selinux-policy
Updated      selinux-policy-targeted-3.9.7-18.fc14.noarch
Cleaned up   selinux-policy-targeted
Session restart required: cronie-1.4.5-4.fc14.x86_64
Please logout and login to complete the update.
```

9.1.6. Available Updates

```
$ pkcon get-updates

Transaction:\tGetting updates
Package:\tbinutils-2.20.51.0.7-6.fc14.x86_64
Package:\ttelepathy-logger-0.1.7-1.fc14.x86_64
Package:\tmdadm-3.1.3-0.git20100804.2.fc14.x86_64
Package:\tlibicu-4.4.1-6.fc14.x86_64
Results:
Bug fix      binutils-2.20.51.0.7-6.fc14.x86_64
Bug fix      libicu-4.4.1-6.fc14.x86_64
Bug fix      mdadm-3.1.3-0.git20100804.2.fc14.x86_64
Bug fix      telepathy-logger-0.1.7-1.fc14.x86_64
```

9.1.7. Groups

Listing Groups

```
$ pkcon get-groups

admin-tools
desktop-gnome
desktop-kde
desktop-other
desktop-xfce
education
```

```

fonts
games
graphics
internet
legacy
localization
multimedia
office
other
programming
publishing
servers
system
virtualization
collections
newest

```

9.1.8. Repositories

Listing Repositories

```

$ pkcon repo-list

Transaction:\tGetting repositories
Status: \tStarting
Results:
  Disabled    rpmfusion-nonfree-updates-testing RPM Fusion for Fedora 14 - Nonfree - Test
Updates
  Disabled    _local                        Automatic local repo. (manged by the "local" yum
plugin).
  Disabled    updates-debuginfo            Fedora 14 - x86_64 - Updates - Debug
  Disabled    rawhide-source               Fedora - Rawhide - Source
  Disabled    updates-testing              Fedora 14 - x86_64 - Test Updates
  Disabled    updates-testing-source       Fedora 14 - Test Updates Source
  Enabled     fedora                       Fedora 14 - x86_64
  Disabled    fedora-latam                 Fedora Latam 14 - x86_64
  Enabled     fedora-chromium              Chromium web browser and deps
  Disabled    updates-source               Fedora 14 - Updates Source
  Disabled    rawhide-debuginfo            Fedora - Rawhide - Debug
  Disabled    fedora-debuginfo             Fedora 14 - x86_64 - Debug
  Disabled    updates-testing-debuginfo     Fedora 14 - x86_64 - Test Updates Debug
  Disabled    rawhide                      Fedora - Rawhide - Developmental packages for the next
Fedora release
  Disabled    fedora-source                Fedora 14 - Source
  Disabled    rpmfusion-free-updates-debuginfo RPM Fusion for Fedora 14 - Free - Updates Debug
  Disabled    rubyera                      Rubyera repo
  Enabled     updates                      Fedora 14 - x86_64 - Updates

```

Enable a Repository

```

$ pkcon repo-enable rubyera

Transaction:\tEnabling repository
Status: \tWaiting for authentication
Status: \tStarting
Status: \tGetting information
Results:

```

Disable a Repository

```
$ pkcon repo-disable rubyera

Transaction:\tEnabling repository
Status: \tWaiting for authentication
Status: \tStarting
Results:
```

9.1.9. Downloading Packages

You must specify a directory where to download, and the package name.

```
$ pkcon download . fwsnort
Transaction:\tDownloading packages
Status: \tStarting
Package:\tfwsnort-1.0.6-8.fc14.noarch
Percentage:\t0
Status: \tDownloading packages
Percentage:\t100
Status: \tCopying files
Percentage:\t70
Percentage:\t100
Results:
Downloaded    fwsnort-1.0.6-8.fc14.noarch
$ ls
fwsnort-1.0.6-8.fc14.noarch.rpm
```

9.1.10. Use of Filters

Listing available Filters

```
$ pkcon get-filters
\t
installed
devel
gui
free
basename
newest
arch
```

Using an example filter

Available updates for GUI graphical applications.

```
$ pkcon --filter=gui get-updates
Transaction:\tGetting updates
Bug fix\tgoffice-0.8.10-1.fc13.x86_64          \tGoffice support libraries
```

9.1.11. Transactions

Listing Transactions

```

$ pkcon get-transactions

Transaction:\tGetting transactions
Results:
Transaction: /5107_dcbacbcc_data
System time: 2010-12-08T07:47:51.024966Z
Succeeded: True
Role: update-system
Duration: 53 (seconds)
Command line: pkcon
User ID: 0
Username: root
Real name: root
Affected packages:
- updating binutils-2.20.51.0.7-6.fc14.x86_64
- updating mdadm-3.1.3-0.git20100804.2.fc14.x86_64
- updating libicu-4.4.1-6.fc14.x86_64
- updating telepathy-logger-0.1.7-1.fc14.x86_64
- cleanup binutils
- cleanup mdadm
- cleanup libicu
- cleanup telepathy-logger
Transaction: /5092_bbcceeab_data
System time: 2010-12-08T07:10:02.594755Z
Succeeded: True
Role: remove-packages
Command line: pkcon
User ID: 0
Username: root
Real name: root
Affected packages:
- removing qt octave
- removing octave
- removing suitesparse
Transaction: /5089_daabbacb_data
System time: 2010-12-08T07:05:05.794247Z
Succeeded: True
Role: update-packages
Duration: 192 (seconds)
Command line: pkcon
User ID: 0
Username: root
Real name: root
Affected packages:
- updating binutils-2.20.51.0.7-6.fc14.x86_64
- cleanup binutils
...

```

9.2. gpk-install-local-file

It install local software package files. It is also executed when it is done a double-click on the software package in your desktop or in your file browser.

```

$ gpk-install-local-file -h
Usage:
  gpk-install-local-file [OPTION...] gpk-install-file

PackageKit File Installer

Help Options:
  -h, --help                Show help options
  --help-all               Show all help options
  --help-debug              Show debugging options

```

```
--help-gtk          Show GTK+ Options

Application Options:
  -v, --verbose      Show debugging information for all files
  --display=DISPLAY  X display to use
```

9.3. gpk-install-catalog

gpk-install-catalog reads a .catalog input file that makes that the packages properly listed to be installed if they are not already. This PackageKit's option can be very useful to deployers and software developers as it allows the user for example to download and install a catalog in his distribution without having to worry about identifying the components.

It is sort of equivalent to yum groups but more specific. What does a .catalog look like?

```
[PackageKit Catalog]

# Easy, the Fedora package which provides the required dependency
# If multiple packages, asks the user
InstallProvides(fedora)=audio/QCELP

# Specific to Fedora 9, installs two development packages
InstallPackages(fedora;9)=glib2-devel;PolicyKit-gnome-devel

# In any distro, installs the package which provides the indicated file
InstallFiles=/usr/bin/fontinst

# For every Fedora 8 arch, installs one of two different compat files
InstallFiles(fedora;8;i686)=/usr/lib/pango/1.6.0/modules/pango-arabic-fc.so
InstallFiles(fedora;8;x64)=/usr/lib64/pango/1.6.0/modules/pango-arabic-fc.so
```

Note the distro-id in (), this allows the packager to spell the package name, eg PolicyKit is the name in Fedora, but policykit is the name in Debian. It can be granular to distribution, distribution version and architecture.

Note that there is no version checking. You must specify the distribution and it must have the correct data.

If there is an optional package not present in earlier versions of the distribution, then you can do something like this:

```
[PackageKit Catalog]

# Fedora 9 does not have Unique
InstallPackages(fedora;9)=glib2-devel;PolicyKit-gnome-devel

# Rawhide is fedora 9.90
InstallPackages(fedora;9.90)=glib2-devel;PolicyKit-gnome-devel;unique-devel
```

There is not a catalog description to translate or check, it is simply a package listing, files, requirements (provides) that may be useful to the user. All translations comes from the distribution metadata and it is up to the user to check the package listing he is asking to install, as there are no signatures or anything like that.

The supported methods are *InstallPackages*, *InstallProvides* and *InstallFiles*. Any line that does not begin with any of these three words will be discarded.

9.4. pkmon

pkmon - PackageKit's Monitor. This application enables real-time view of the progress of PackageKit ongoing transactions.

```
$ pkmon -h

Usage:
  pkmon [OPTION...]

PackageKit Monitor

Help Options:
  -h, --help                Show help options
  --help-all               Show all help options
  --help-debug              Show debugging options

Application Options:
  --version                 Show the program version and exit
  -v, --verbose             Show debugging information for all files
```

```
$ pkmon
Transactions:
[none]
network status=offline
Transactions:
1\t/2923_dadbdeab_data
Transactions:
1\t/2923_dadbdeab_data
2\t/2925_edbcdebb_data
Transactions:
1\t/2923_dadbdeab_data
2\t/2925_edbcdebb_data
3\t/2926_aaaaedac_data
Transactions:
1\t/2923_dadbdeab_data
2\t/2925_edbcdebb_data
3\t/2926_aaaaedac_data
4\t/2927_abcbdcad_data
/2923_dadbdeab_data\tpercentage -1
/2923_dadbdeab_data\trole      simulate-install-packages
/2923_dadbdeab_data\tstatus    setup
/2925_edbcdebb_data\tpercentage -1
/2925_edbcdebb_data\trole      resolve
/2925_edbcdebb_data\tstatus    wait
/2925_edbcdebb_data\tallow_cancel 1
/2926_aaaaedac_data\tpercentage -1
/2926_aaaaedac_data\trole      get-distro-upgrades
/2926_aaaaedac_data\tstatus    wait
/2926_aaaaedac_data\tallow_cancel 1
/2927_abcbdcad_data\tpercentage -1
/2927_abcbdcad_data\trole      get-updates
/2927_abcbdcad_data\tstatus    wait
/2927_abcbdcad_data\tallow_cancel 1
```


Rpm management elements

As the reader will have realized throughout this manual, *YUM* and *PackageKit* can resolve most of the systems administrator needs when managing software for Fedora systems. There remains some edge cases where you can not have a satisfactory answer at high level, with *YUM* and *PackageKit*. For such cases the lower level tool **rpm** must be used and you must understand how crucial it is. This section does not attempt to give complete information of *RPM* system operation or all of its configuration options, but to offer a complementary survival introductory guide to *RPM* in order to avoid the read of the full manual for at least the following cases.

10.1. Why avoid RPM?

In this chapter introduction it has not been said why an administrator should avoid using **rpm** in the daily software management. One of the reason is because it is a "low level" tool, which means that upper level tools, *YUM* and *PackageKit*, are not notified about the use of **rpm**. *YUM* will notify the user about a change to the rpm database made by an foreign application.

Generally you will want to avoid this type of messages and keep software management as simple and unified as possible. Of course, you may always find certain unresolvable cases in which you will have no choice but to use *rpm* to solve the problem or particular case.



It will be manifested as an interference

yum will log and output as an interference the access to *RPM* system.

```
$ yum install ...\t\t\t\t
\t\t\t\t...
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
Warning: RPMDB altered outside of yum.
...
```

10.2. Documentation in the package

If the Packers have been diligent, **rpm** can list files previously tagged as *documentation*:

```
$ rpm -qd mc
/usr/share/doc/mc-4.6.1a/COPYING
/usr/share/doc/mc-4.6.1a/FAQ
/usr/share/doc/mc-4.6.1a/NEWS
/usr/share/doc/mc-4.6.1a/README
/usr/share/man/es/man1/mc.1.gz
/usr/share/man/hu/man1/mc.1.gz
/usr/share/man/it/man1/mc.1.gz
/usr/share/man/man1/mc.1.gz
/usr/share/man/man1/mcedit.1.gz
/usr/share/man/man1/mcview.1.gz
/usr/share/man/pl/man1/mc.1.gz
/usr/share/man/ru/man1/mc.1.gz
/usr/share/man/sr/man1/mc.1.gz
```

10.3. Configuration files in the package

Continuing the above example, to know what configuration files are included in the package *mc* the query option **-c** must be used:

```
$ rpm -qc mc
/etc/mc/cedit.menu
/etc/mc/edit.indent.rc
/etc/mc/edit.spell.rc
/etc/mc/extfs/extfs.ini
/etc/mc/extfs/sfs.ini
/etc/mc/mc.charsets
/etc/mc/mc.ext
/etc/mc/mc.lib
/etc/mc/mc.menu
/etc/mc/syntax/Syntax
```

10.4. File ownership

When a rpm package is installed it declares ownership over the files it installs. Later it's possible to query back the database, in order to find the package owner of a given file, an example below:

```
$ rpm -qf /usr/bin/xwininfo
xorg-x11-utils-7.4-9.fc13.x86_64
```

10.5. Database backup and rebuild

It may happen that the RPM database is corrupted for various reasons. If so, you can rebuild the entire database just having **/var/lib/rpm/Packages** in a good condition, the other files are generated from it, and please, backup your database often.

Later, to restore the RPM database from the file **Packages**:

```
$ su -
# cd /var/lib/rpm
# rm __db.*
# cp /root/Packages.bu Packages
# rpm --rebuilddb
```

10.6. Importing keys

To verify the rpm packages you must import the public keys from the packager. To do this run the following command at the command line:

```
$ rpm --import /ruta/al/archivo/con/la/llavepki
```

10.7. List imported keys

```
$ rpm -qa gpg-pubkey*
```

```

gpg-pubkey-536ceb2a-419e7fcd
gpg-pubkey-2425b284-4affedb7
gpg-pubkey-66534c2b-495ca3b8
gpg-pubkey-a3780952-4affed06
gpg-pubkey-e8e40fde-4b563cdc
gpg-pubkey-a109b1ec-3f6e28d5
gpg-pubkey-16ca1a56-4a100959
gpg-pubkey-57bbccba-4a6f97af
gpg-pubkey-f6777c67-45e5b1b9
gpg-pubkey-a7a1d4b6-43281558
gpg-pubkey-a3a882c1-4a1009ef

```

For information on a specific key:

```

$ rpm -qi gpg-pubkey-2425b284-4affedb7
Name       : gpg-pubkey                      Relocations: (not relocatable)
Version    : 2425b284                        Vendor: (none)
Release    : 4affedb7                        Build Date: vie 28 may 2010 05:56:16 VET
Install Date: vie 28 may 2010 05:56:16 VET   Build Host: localhost
Group      : Public Keys                     Source RPM: (none)
Size       : 0                               License: pubkey
Signature  : (none)
Summary    : gpg(RPM Fusion nonfree repository for Fedora (13) <rpmfusion-
buildsys@lists.rpmfusion.org>)
Description :
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: rpm-4.8.0 (NSS-3)

mQINBer/7bcBEADR9g8j2q5iI+NUJ6ZHS5yht56nPP4I9US/HcPs6PsozMtT0H3j
eLA+nH6DSn9rVUTS84anaUB6uSf/yzd5yUibnw4XZ6ZE7MaEjM/GIRymRqkV7sIX
ZTDU88mv3KIHbjC611iR0fkH7l8Dl3aWRW0p1XY/G8LixVC5ExzSFHRI02VXo6nM
UpnxEoysyvcVsm+gmnRYDnwN7edYt0M4vHeRcedKu8uhzqw2dBwiSzZssCIZR7/
qeJEqwSzuEjf7hN7iGIhjQcfrNCxHdcGk8/1tZL2XnDZe/Uxe28nfUTE5wGufY
Y70sr3Bnx0eJc/nsGH7RHSBSII2NNXZPmvCSMx0wfNUXq37kok8XkUwL+crV/YIw
7lcUWTEdy1Y3VAt01uF5EXdrWrSjhmArVM0xiGDYCuYyABZw84t1I0wq/PKeHYuy
nDK6JNGa+ZN9Hr/houp7wqxoX0v6SfnAvTGKH0QASsAiPQQzDDZReT+rDw+PKmD
Y+TYVMDoh/hafSFmG0AjShoIl/1PBwy+U0ncMeImvwB0kMR95ygheFe29hDYJH06
hiX6H9kBTztvImsKwEw0iPozA1UzdknvyP+2cHco0Clg+uZFeq7aFJ/E5Fw72R09
iy8YFeQoN84QAQp0m1rl3GGnGFRLPEGGi92KPXUvvHvaP+cqjGPnFtBhbQARAQAB
tFZSUE0gRnVzaW9uIG5vbWZyZWUgcmlvbmV3NpdG9yeSBmb3JlICGxMykg
PHJwbWZ1c2lvdWl5ZHN5c0BsaXN0cy5ycG1mdXNpb24ub3JnPokCNgQTAQIA
IAUCsv/ttwIbAwYLCQgHAwIEFQIIAwQWAgMBAh4BAheAAAJECvVL9IkJbKEGksQ
AIAK65TxlFj+F4JjRFyd1YvHEubJKoTpgZ9r4Sz7hcVD0dvnesHugcERvzGrDgd4
XrmKd0bojBHPTQYR0UQ6tu85KdYABZ0QT0uB6vKxXVbZgvfNqJzuqsMUKlma6uGo
L0Cy9W6H3YmzmtaupuxCW5HBvWzN30jXH7KQmw4enH8+2qy1p1mDSHyFuBjOF6UE
8Jf0q9dsjYArTrTbGF16n8vrdGCBKHXhGW9J8gsy0IXo8yZzYEkytUjydtR3R/9
A8LpLeHniPGU/hnNPHVYZPnYcT7XLY8uhwTgBvhglCVlp9ouZjhPomfqw/qNmjWh
zDY7lYtEIKeywweOGw9Ft/WCgFlB7X3V4OLK3P2c9HGoAQ1WAohU5Y8gpqyrSt0d
fs7Pr1HoZbYG7bEPc3x9VXAcnMc+FGYP7X1J/1NXWNVqaBrBp8dB73FCNHJOGfa8
x25V8Tnq1aqCWus+BQQCSdz637PZrx6KjcTRRLuiEU1KiNeCgFJP6jyffzkbP1P1
PNTTrD76qoQfNPZqQKlDTwcIii6Yo87SW0vXBZvokrArPxWTFs9fcuHXBS2axHg3
cLRUC03TASCDsZ6WtygouphUBAtDgSC71DL1CbIN8EmXN6C0sSB5Ywnzdsy7yIT2
2yrVZna0hI/WBR+AQw8jszgJ5JVdEDftqSAKBUXXKela
=3/K0
-----END PGP PUBLIC KEY BLOCK-----

```

Techniques for system administrators

The following are introduction to additional topics for advanced users rather than the material that has been presented so far in this manual and may be useful in more complex software management environments, in particular, Fedora desktops or servers network, computer centers, etc.

11.1. Remote configuration

It is possible to centralize all **yum** configuration on one external server and make **yum** obtain those settings at runtime. Later, the administrator can even have multiple settings for the same server and given the specific circumstances, use one or another setting at will.

The first useful option in this direction is the execution of **yum --config (-c)** command, which lets you specify the configuration file to use, that in Fedora defaults to **/etc/yum.conf**, however, you can use HTTP and FTP URLs as the remote location for the configuration file.

```
# yum --config=http://central.repo.conf/yum.conf
```

If it is combined with a **bash** alias, results in a clean solution for the local user.

```
$ alias yum='yum --config=http://central.repo.conf/yum.conf '
$ alias
alias l.='ls -d .* --color=auto'
alias ll='ls -l --color=auto'
alias ls='ls --color=auto'
alias vi='vim'
alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot --show-tilde'
alias yum='yum --config=http://central.repo.conf/yum.conf '
```

The syntax *include* for inclusion in **/etc/yum.conf** and **.repo** files in **/etc/yum.repos.d/** allows the addition of additional remote settings via an URL, and this is the second way to load remote configuration. Use the format:

```
include=url://path/to/the/configuration/file
```

The configuration file will be inserted in the position of *include*. The file inclusion can be nested with new *include*. **yum** will abort when an infinite loop error is detected.

11.2. syslog support

The following three options allow to configure **yum** syslog behavior.

syslog_ident

Sets the id, program name to be use for syslog messages.

syslog_facility

Sets the "facility" name for syslog messages generated by **yum**, see man syslog(3). It defaults to 'LOG_USER'.

syslog_device

Tells **yum** where to send syslog messages. It may be a local device (path) or a string of the form computer:port to use a remote syslog system. If it is not defined, or if it points to a nonexistent device, then syslog is disabled. By default, it is **/dev/log**.

Displays YUM syslog messages in Fedora.

```
# grep yum /var/log/messages | tail
Oct  1 16:20:33 movix yum[10917]: Installed: ruby-dbus-0.3.0-1.fc13.noarch
Oct  1 17:40:45 movix yum[12154]: Installed: ruby-cairo-1.8.5-1.fc13.x86_64
Oct  1 17:40:47 movix yum[12154]: Installed: ruby-pango-0.19.4-2.fc13.x86_64
Oct  1 17:40:49 movix yum[12154]: Installed: ruby-gdkpixbuf2-0.19.4-2.fc13.x86_64
Oct  1 17:40:51 movix yum[12154]: Installed: ruby-libart2-0.19.4-2.fc13.x86_64
Oct  1 17:40:53 movix yum[12154]: Installed: ruby-atk-0.19.4-2.fc13.x86_64
Oct  1 17:40:58 movix yum[12154]: Installed: ruby-gtk2-0.19.4-2.fc13.x86_64
Oct  1 17:41:00 movix yum[12154]: Installed: ruby-gnomecanvas2-0.19.4-2.fc13.x86_64
Oct  1 17:41:02 movix yum[12154]: Installed: ruby-gnome2-0.19.4-2.fc13.x86_64
Oct  2 07:27:22 movix yum[16700]: Installed: fwsnort-1.0.6-7.fc13.noarch
```

Appendix A. Revision History

Revision 0 **Thu Sep 20 2010**

Guillermo J. Gómez Savino
[*guillermo.gomez@gmail.com*](mailto:guillermo.gomez@gmail.com)

Initial creation of book by publican

Index

F

feedback

contact information for this manual, xi
