Mastering Linux, part 9

This month *Jarrod Spiga* demonstrates how to install packages on a Linux system with Red Hat Package Manager (RPM) and Yellow dog Updater, Modified (yum).

On the cover DVD

Bonus DVD software

PDFs of every instalment of the Mastering Linux series.

Skill level

Intermediate

Requirements

An installation of Linux (Fedora Core 3 was used in the writing of this article) • a working network or Internet connection.

Time to complete

3 hours (approx)

Why spend hundreds or even thousands of dollars on fancy boxed software packages when there are often open source equivalents readily available for little or no cost? For the most part, finding the software is a piece of cake. Most hurdles are encountered when it comes to installing packages, since the process is a little more involved than running setup.exe. It can be a bit difficult getting your head around managing prerequisite, co-requisite and conflicting packages, let alone compiling your own applications. But there are tools available to make things easier.

ADD AND REMOVE PROGRAMS

Part of the reason why Fedora Core 3 is a four-CD download is that it ships with an extensive array of popular software packages. Most aren't automatically installed to conserve disk space, but all are readily accessible from the CDs should you need them.

1 The quickest and easiest way to check what's on the CDs is to use the Add and Remove Software tool, located under GNOME > System Settings. If you aren't logged in as the root user, you'll be asked for the root password to continue.

At launch, the tool scans your system and compiles a list of the software that's already installed. Then the Add Package Groups screen appears.

To add or remove packages, drill down into the relevant group by clicking on Details. Ensure all checkboxes for the applications you want to install are ticked.

Some packages within program groups won't have checkboxes next to them. These are usually prerequisite packages which must be installed for that program group.

After selecting the packages you want to install, hit Install Packages at the bottom of the Add and Remove Software tool. Confirm your selections at the dialog to begin the install. You may be asked to insert the installation media into your optical drive during the procedure.

2 After adding the packages, run up2date and download any updates available for the newly-installed applications. The sole drawback to installing from CD is that you don't always have the latest, most secure versions.

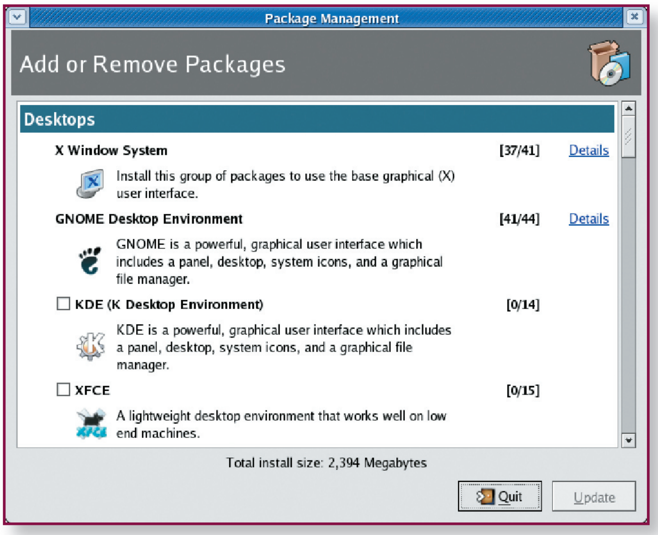
A TASTE OF YUM

The Add and Remove Software tool only lists the packages bundled on the Fedora Core 3 CDs. There are hundreds of other applications downloadable via the Yellow dog Updater, Modified (yum).

Yum is an automated update and package installation app for RPM-based systems (including Fedora) that runs at the command line. Its main purpose is to determine all the dependencies for packages being installed or upgraded, and to figure out which order to install those dependencies so everything works.

Yum is the application used to download and install updates in the correct order.

▶



Add or Remove Packages

Desktops

X Window System [37/41] Details
Install this group of packages to use the base graphical (X) user interface.

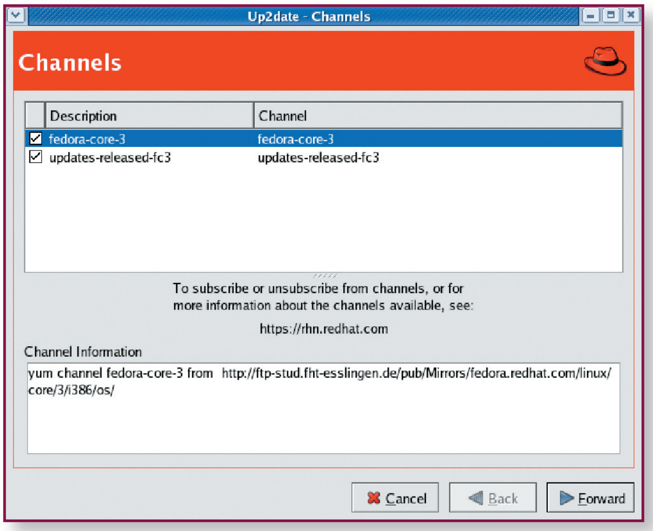
GNOME Desktop Environment [41/44] Details
GNOME is a powerful, graphical user interface which includes a panel, desktop, system icons, and a graphical file manager.

□ KDE (K Desktop Environment) [0/14]
KDE is a powerful, graphical user interface which includes a panel, desktop, system icons, and a graphical file manager.

□ XFCE [0/15]
A lightweight desktop environment that works well on low end machines.

Total install size: 2,394 Megabytes

1 ***Pick a package:*** *many software applications can be installed on your Linux system using the Add and Remove Packages tool.*



Channels

Description	Channel
☑ fedora-core-3	fedora-core-3
☑ updates-released-fc3	updates-released-fc3

To subscribe or unsubscribe from channels, or for more information about the channels available, see:

https://rhn.redhat.com

Channel Information

```
yum channel fedora-core-3 from   http://ftp-stud.fht-esslingen.de/pub/Mirrors/fedora.redhat.com/linux/
core/3/i386/os/
```

2 ***Always update:*** *after installing packages from CD, it's a good idea to run up2date to ensure that those are the latest stable versions.*

▶ Configuring yum was explained in part one of this series (*APC* December 2004, page 98; also in PDF format on this month's cover DVD).

3 To download and install software using yum, you'll need to know the name of the package you want. You can use the `list` argument to the `yum` command to guess package names, but it can take some time for online repositories to tell you whether you've guessed correctly. A better way is to create a single file with a listing of all of the packages available. To do so, bring up a command prompt and run as the root user:

```
yum list available > ~/yum-list.txt
```

To search through a list of packages, simply run the following command:

```
cat yum-list.txt | grep <name>
```

where `<name>` is part of the name of the packages you're looking for.

The advantage of yum is that you'll always download and install the latest version of the package — there's no need to update after installation. And when future versions come out, you don't have to do anything more than run up2date again.

RPM

Even though the yum repository contains thousands of software packages, you may come across an application that's not available for download via yum.

Many developers package software for download in RPM format — the same format as yum. However, installing an RPM is more involved than using yum: you have to download all the required packages and resolve any dependencies or conflicts manually before installation.

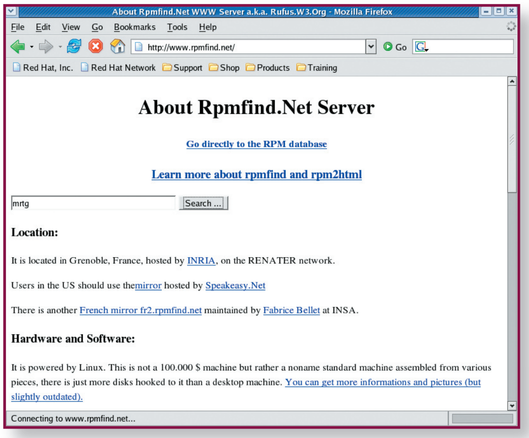
4 Installing and upgrading RPM packages from the GUI is very simple: just navigate to the location of the appropriate RPM package and double-click on it. If you're not logged in as root, you'll be prompted for the root password to proceed. Your system will then check what prerequisites are needed.

5 Once this evaluation is complete, a dialog will appear that reports how many packages will be installed, and how much disk space they will consume. Select continue and Fedora installs or upgrades everything, including the prerequisite RPM files (if Fedora can find them).

If Fedora can't find any required RPM files, an error message pops up displaying the packages that can't be located. You'll need to resolve any failed dependencies (see "Failed dependencies", right) before the installation will succeed.

You can also install RPM packages from the command line. The `rpm` command can perform a variety of tasks — running `rpm --help` from the command line lists the functions available.

The following command line can be used to install a package via RPM:



About Rpmfind.Net WWW Server a.k.a. Rufus.W3.Org - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.rpmfind.net/
Go
Red Hat, Inc.
Red Hat Network
Support
Shop
Products
Training

About Rpmfind.Net Server

Go directly to the RPM database

Learn more about rpmfind and rpm2html

mrtg Search ...

Location:

It is located in Grenoble, France, hosted by INRIA, on the RENATER network.

Users in the US should use themirror hosted by Speakeasy.Net

There is another French mirror fr2.rpmfind.net maintained by Fabrice Bellet at INSA.

Hardware and Software:

It is powered by Linux. This is not a 100.000 $ machine but rather a noname standard machine assembled from various pieces, there is just more disks hooked to it than a desktop machine. You can get more informations and pictures (but slightly outdated).

Connecting to www.rpmfind.net...

Package hunt: *Rpmfind.net is an array of mirrors containing thousands of applications packaged in RPM format, ready for your Fedora Core 3 system.*

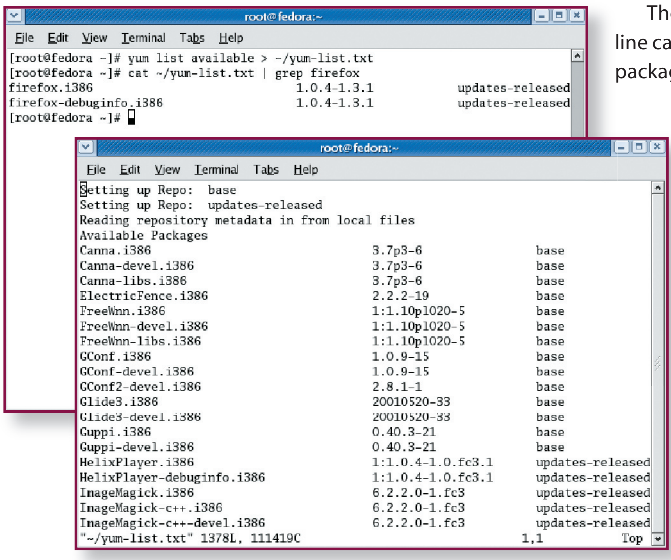
```
rpm -Ivh <package name>
```

where `<package name>` is the name of the package(s) that you're installing. In this example, the `v` switch to the rpm command provides more detailed output. Without it, the rpm command only displays errors encountered. The `h` switch prints hashes (#) as a pseudo-progress bar at the command line, giving you an idea of how the installation is proceeding.

A similar command can be used to upgrade packages, though you'd use the `U` (for upgrade) switch instead of `I` (for install).

FAILED DEPENDENCIES

Managing dependencies is one of the more difficult tasks when installing RPM packages. In extreme cases, you'll need to satisfy a number of prerequisites to install a package, but some of those prerequisites have prerequisites of their own. ▶



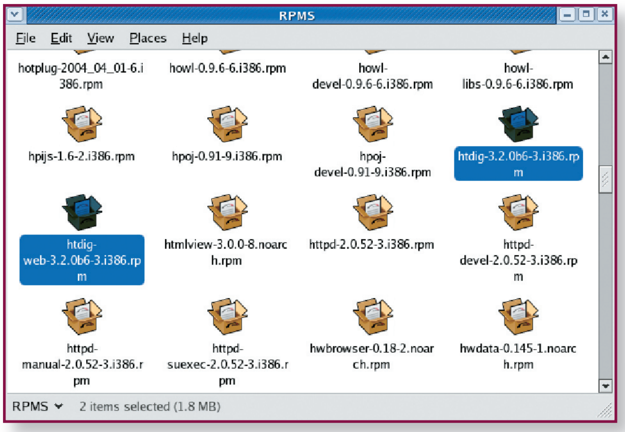
root@fedora:~
File Edit View Terminal Tabs Help

```
[root@fedora ~]# yum list available > ~/yum-list.txt
[root@fedora ~]# cat ~/yum-list.txt | grep firefox
firefox.i386                            1.0.4-1.3.1            updates-released
firefox-debuginfo.i386                  1.0.4-1.3.1            updates-released
[root@fedora ~]# 
```

root@fedora:~
File Edit View Terminal Tabs Help

```
Setting up Repo:  base
Setting up Repo:  updates-released
Reading repository metadata in from local files
Available Packages
Canna.i386                              3.7p3-6                base
Canna-devel.i386                        3.7p3-6                base
Canna-libs.i386                         3.7p3-6                base
ElectricFence.i386                      2.2.2-19               base
FreeWnn.i386                            1:1.10pl020-5          base
FreeWnn-devel.i386                      1:1.10pl020-5          base
FreeWnn-libs.i386                       1:1.10pl020-5          base
GConf.i386                              1.0.9-15               base
GConf-devel.i386                        1.0.9-15               base
GConf2-devel.i386                       2.8.1-1                base
Glide3.i386                             20010520-33            base
Glide3-devel.i386                       20010520-33            base
Guppi.i386                              0.40.3-21              base
Guppi-devel.i386                        0.40.3-21              base
HelixPlayer.i386                        1:1.0.4-1.0.fc3.1      updates-released
HelixPlayer-debuginfo.i386              1:1.0.4-1.0.fc3.1      updates-released
ImageMagick.i386                        6.2.2.0-1.fc3          updates-released
ImageMagick-c++.i386                    6.2.2.0-1.fc3          updates-released
ImageMagick-c++-devel.i386              6.2.2.0-1.fc3          updates-released
"~/yum-list.txt" 1378L, 111419C                          1,1           Top
```

3 ***List all that's yum:*** *saving a list of all available yum packages makes for speedy searching in the future.*



RPMS
File Edit View Places Help
hotplug-2004_04_01-6.i386.rpm
howl-0.9.6-6.i386.rpm
howl-devel-0.9.6-6.i386.rpm
howl-libs-0.9.6-6.i386.rpm
hpijs-1.6-2.i386.rpm
hpoj-0.91-9.i386.rpm
hpoj-devel-0.91-9.i386.rpm
htdig-3.2.0b6-3.i386.rpm
htdig-web-3.2.0b6-3.i386.rpm
htmlview-3.0.0-8.noarch.rpm
httpd-2.0.52-3.i386.rpm
httpd-devel-2.0.52-3.i386.rpm
httpd-manual-2.0.52-3.i386.rpm
httpd-suexec-2.0.52-3.i386.rpm
hwbrowser-0.18-2.noarch.rpm
hwdata-0.145-1.noarch.rpm
RPMS
2 items selected (1.8 MB)

4 ***Quick and easy:*** *installing RPM packages is too easy when using the GUI — provided that X-Windows can find all the required packages.*

If you have failed dependencies, the first place to look is on the Fedora CDs. Insert the first Fedora Core 3 CD and at a command line, mount the disk by entering:

```
mount /mnt/cdrom
```

Search for the required RPM by typing:

```
find /mnt/cdrom -print | grep
<dependency>
```

where `<dependency>` is the name of the dependency you're looking for. If you find it, copy the relevant RPM files to the same location as the RPM. If you don't find the dependency, unmount the CD-ROM:

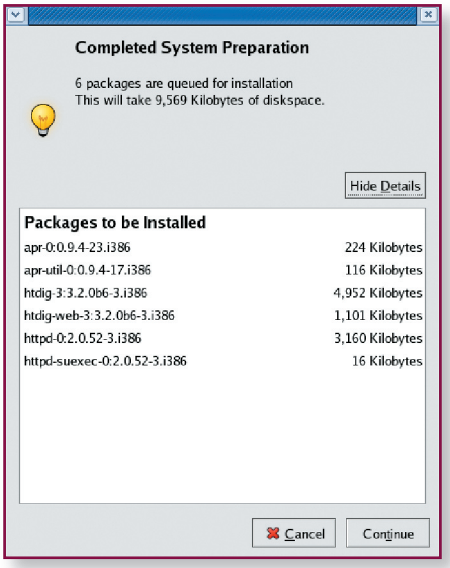
```
unmount /mnt/cdrom
```

and search the next CD. If you've searched every CD and still can't find it, there are a number of online RPM libraries that can be searched, such as **www.rpmfind.net**.

CONFLICTS

Occasionally, you may encounter a conflict when installing or upgrading software. For instance, RPM might inform you that if you install a new version of Perl, a number of already installed applications won't work properly with the newer version.

Conflicts such as this can be very difficult to solve. Generally, you'll need to source updated versions of all affected packages in order to get the installation to proceed.



Completed System Preparation

6 packages are queued for installation
This will take 9,569 Kilobytes of diskspace.

Packages to be Installed

apr-0:0.9.4-23.i386	224 Kilobytes
apr-util-0:0.9.4-17.i386	116 Kilobytes
htdig-3:3.2.0b6-3.i386	4,952 Kilobytes
htdig-web-3:3.2.0b6-3.i386	1,101 Kilobytes
httpd-0:2.0.52-3.i386	3,160 Kilobytes
httpd-suexec-0:2.0.52-3.i386	16 Kilobytes

5 ***Automatic installs:*** *when installing RPMs from the GUI, any dependencies that aren't installed are automatically added to the installation list.*

BUILDING AND COMPILING

Almost every application under Linux can be compiled directly from source code, even if RPMs are available. Most applications for Linux are written using the C programming language, and the GNU Compiler Collection (GCC) is used to build and compile them.

This process usually involves three steps:

- **Configuration** making any customisations and applying any settings to the package.
- **Building** the actual compilation of the source code and your configuration to create the executable files and data that make up the application.
- **Installation** copying the executable files and data to appropriate locations on your file system.

You'll usually download the source code in a gzipped tarball format — a format similar to a ZIP file and carries the TAR.GZ extension. To extract the source file from the gzipped tarball, you could run:

```
tar -zxvf <name>
```

where `<name>` is the name of the downloaded gzipped tarball.

The source code for most apps contains a file called INSTALL, which usually contains specific instructions on how to build that application. In general, you can configure the application by switching to the directory where the source code is located and entering:

```
./config
```

A configuration script should run, perhaps prompting for certain pieces of information necessary for compilation. After configuration, build the application by running:

```
make
```

Don't expect to understand a lot of the output — it's often useful in debugging a compilation, but will only be of use should your build fail.

Once your system stops processing code, look at the last few lines of output. If you see any errors, the build has failed. You'll need to rectify these errors before running the make command again. In most cases, failed builds are the result of missing prerequisites or a bad configuration. In both cases, refer to the INSTALL file.

GET GOING

Once you have a successful build, copy the necessary files to working locations by typing:

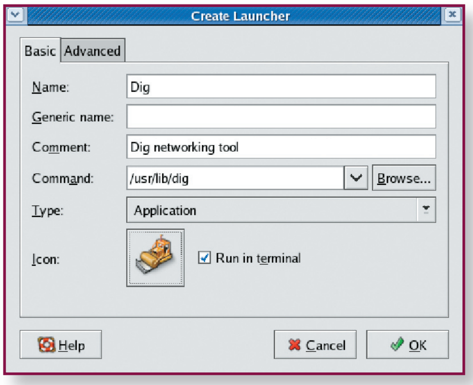
```
make installStart me up
```

When you install an application in Windows, the installation routine usually creates a new program group and icons for your application in the Start menu. In Linux, you'll need to create an icon under the GNOME menu.

6 Making an application launcher is easy. Simply right-click in an empty area of the taskbar and choose Add to Panel > Custom Application Launcher. A dialog box appears in which you can enter in the details of the appropriate app.

The Name and Command fields must be completed in order to create the button: the latter must contain the full path to the application (the Browse button may help you fill in this detail correctly). Alternatively, enter a command line argument in this field to alter the way the app behaves. Configure an icon for the application if required.

Naturally, the Run in terminal checkbox should be selected if you're using a command line app — you won't see much output if you don't. Click OK and your new launcher should be ready to use.



Name: Dig
Generic name:
Comment: Dig networking tool
Command: /usr/lib/dig
Type: Application
Icon:
☑ Run in terminal

6 ***Icon do that:*** *Linux installers generally don't add program groups to X-Windows. You'll need to create launchers to start applications at the click of a button.*

Next month . . .

Next month's instalment of Mastering Linux takes you back to the shell and explains some of the more powerful commands. It also guides you through the process of creating your first shell scripts.