



The fast and sensible Smart package manager

# GET SMART

The package manager is one of the central components on any Linux system. If you have a system with unreliable package management – such as Suse Linux 10.1 – you may want to consider the Smart alternative.

BY MARCEL HILZINGER

**T**he true test of a package manager is how the tool resolves dependencies. Smart is one of the best package managers around for navigating through dependency issues. The Smart package manager does not depend on any specific distribution, and it can handle more than ten repository formats. Smart comes with the *smart* command line tool, the Smart shell (*smart -shell*), and a graphical user interface (Figure 1), which you can launch by entering *smart --gui*.

## How Smart?

The goal of the Smart project is to create “smart and portable algorithms for solving adequately the problem of managing software upgrade and installation.”

There is a reason why this description from the the Smart developers emphasizes the algorithms. Algorithms are an

important focus of the Smart project -- and an important reason why the Smart developers believe their tool is different. When it comes to dependencies, Smart does not just use the easiest and most obvious solution but actually weighs every possible option using a policy-based priority system. In some cases, the “best” solution may depend upon the goal. For instance, if you are performing an upgrade, Smart may not choose the most recent version of a package if an intermediate version provides a better result for the complete system. Smart algorithms also consider efficiency factors, choosing the combination of packages that leads to the fastest and most reliable installation.

Smart can manage DEB, RPM, and Slackware packages, and Smart supports APT, YUM, and URPMI repositories. The Smart architecture even provides a

means for adding new package and repository formats.

## Installation

The easiest way to install Smart is to use your existing package manager, assuming you have an Internet connection. Ubuntu users can run the *sudo apt-get install smartpm* command to install the package from the Universe repository. Users with Fedora Core 5 will find Smart in the *extras* repository; just type *yum install smart* to install. Readers with Suse Linux will need to add the Guru repository [2] as an installation source. You can then install Smart with YaST. The project page [1] has details on Smart repositories for other distributions.

To install from the source code, you must have the developer packages for the RPM or Deb system and a number of Python packages. Once you have these

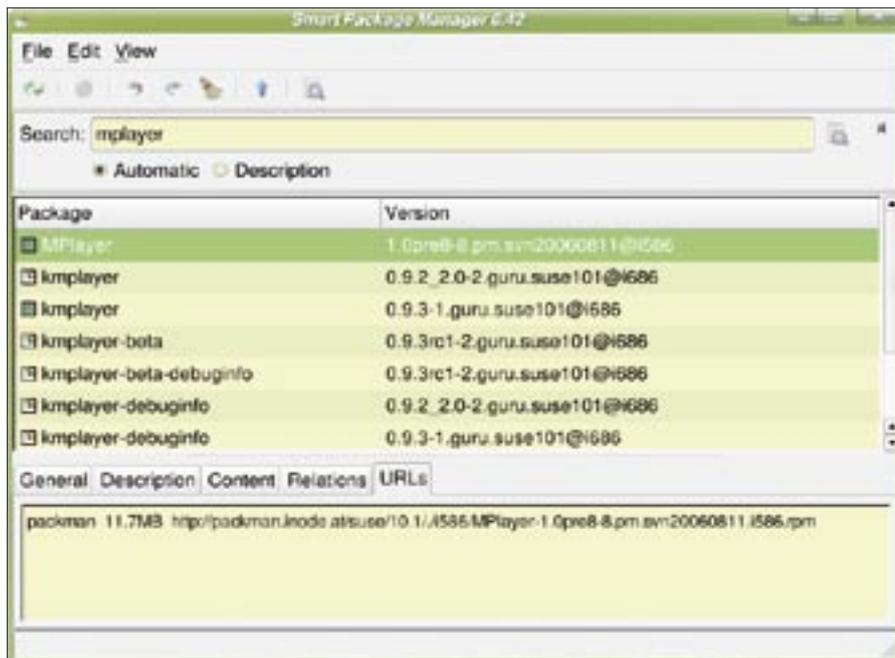


Figure 1: The Smart GUI gives administrators a convenient search function. The URLs tab additionally shows you the repository from which a package originated.

in place, just enter *make* in the source code directory to launch the build, then become root and enter *make install* to install the software.

After completing the install, you can opt to use the command line package manager, as in *smart install amarok* for example, or give the *smart --gui* command to launch the front-end. The software will also support combinations of the two. You need administrative privileges to install or remove software.

## Setting up Repositories

Depending on your distribution and the package provider, Smart may have a channel list that suits your needs. Channels in the Smart sense of the word are any kind of package source or repository. For example, the Suse packages from Guru [3] automatically set up channels for Packman and Guru. Other distributions do not make heavy use of third

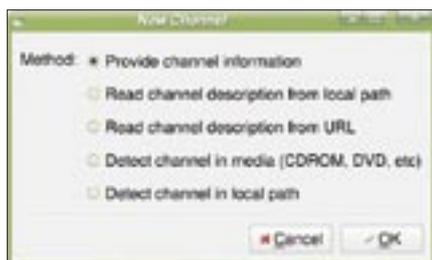


Figure 2: Smart gives you five different methods of integrating a new package source.

party repositories, and very few lists are available. To add a new channel to the GUI, select *Edit | Channel | New*, and then opt for one of the five suggested methods at the next step (Figure 2).

To add a channel manually, select *Provide Channel Information*. To read the RPM packages in a local directory, select the last item. The other three items read existing channel information. The rest of this article explains how to manually set up a repository via *Provide Channel Information*. As an alternative, you can add package sources in the command

### Smart Commands

Install package *smart install package name> smart install /local/path/package name*

Find package *smart search package name*

Update package list (run automatically prior to installation) *smart update*

Add channel *smart channel --add channel file smart channel --add Aliasname name="detailed description" type=apt-deb distribution=dapper baseurl=http://archive.ubuntu.com/ubuntu components="universe"*

Manage mirror servers *smart mirror --add ftp://mainserver.url/path http://mirror.url/path/ smart mirror --show*

Set priorities *smart channel --set Aliasname priority=50*

line by entering *smart channel --add*. The box labeled "Smart Commands" gives you an example for Ubuntu 6.06.

You select the required format in the dialog shown in Figure 3. This is typically *APT-DEB Repository* for Ubuntu and *RPM MetaData* for Fedora Core and Suse Linux. The ability to manage YaST installation sources was added in version 0.42. The *RPM Installed Packages* and *DPKG Installed Packages* entries tell Smart to parse the installed packages, and this gives the tool the ability to delete installed software.

After opting for a format, enter the required information in the *Edit Channel* dialog (Figure 4). As an *Alias*, select a short and intuitive name. You can add a more detailed description in the *Name* box. The *Manual updates* checkbox lets you specify whether Smart should parse the channel when launched. If you enable the checkbox, the tool will leave this step out for the current source. This is useful for static package sources, such as RPM directories on CD, or for local repositories.

Checking *Disabled* tells Smart not to install any packages from this channel. This is practical if you need to disable a channel temporarily without removing the corresponding entry. The *Removable* option tells Smart that the channel is located on an external medium (CD/DVD). If necessary, the package manager will then prompt you to insert the disk.

You can use the *Priority* value to weight channels. The higher the value, the more likely it is that Smart will install a package from this source. You can theoretically assign priorities between *-100000* and *100000*, although a two-figure value should be high enough in most cases. Channels default to a priority of 0.

Specify the download address in the *Base URL* box. An existing configuration file, such as *source.list* on Ubuntu/Debian, or *yum.conf*, is the best source of this information. Users with Fedora Core will need to add entries such as *\$basearch (i386 for example)*.

### Tip

It makes sense to assign highest priorities to your distributor's official package sources, rather than choosing a third-party provider. This avoids issues caused by mixing packages.

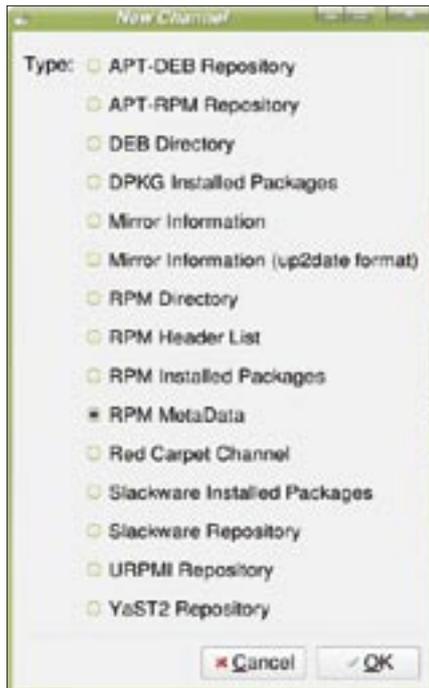


Figure 3: Smart supports a large number of package source formats.

The other fields depend on the selected format. The program will ask you for the *Distribution* and *Components* for Apt-Deb sources. After making changes, make sure you quit Smart by selecting *File | Quit*. In our lab, the program sometimes forgot about new channels if we failed to do so.

## Using Smart

You can press [Ctrl] + [F] or click the magnifying glass icon to enable Smart's most important tool, the search function. By default, the program will only search package names. If you would prefer to search against descriptions, just press the *Description* button. The Smart search function is not case-sensitive, and the command line version of Smart is far more tolerant than *apt-get* in this respect. For example, *smart search realpla* not only finds the RealPlayer package, but all packages with RealPlayer in their descriptions. The tool also chimes in with useful suggestions during the install (see the "Tolerant Search" box).

After completing a search, Smart will first show you the package groups containing the packages. If you prefer to sort the results by package source, select *View | Tree Style | Channel*, although the *View | Tree Style | None* might be more efficient, as it tells Smart to just display the results.

If you are familiar with Synaptic, you should have no trouble using Smart. Clicking a box to the left of the package name tags the software for installation. And you can click to select installed packages for deletion. Yellow stars tell you that updates are available for the packages in question. Right-clicking drops down a menu with advanced functions, such as reinstalling packages, protecting individual packages against updating (*Lock this version*), or setting package priorities.

To install a package, click the cog-wheel icon, select *File | Execute Changes*, or press [Ctrl] + [C]. Smart will attempt a parallel download of the selected packages, accessing various mirrors to do so. The tool remembers the fastest and most reliable mirrors and uses them by preference for future downloads. To add a new mirror, select *Edit | Mirror*, select the main server, and then click *New*. Check out the "Smart Commands" box to find out how to add mirrors.

## Tips and Tricks

Smart can also handle local packages. The *smart install ~/\*.rpm* command installs all the RPM packages, as well as the required channel files. In your home directories. If you would like to convert your *sources.list* into a Smart channel file, see the scripts at [4].

Priorities can be assigned to packages besides channels. This is a good way of preventing updates to a working Xine configuration, for example. However, you will need to think about which packages to assign priorities to and remember your choices, as this may prevent updates to important packages. If in doubt, you might prefer to use *Lock*

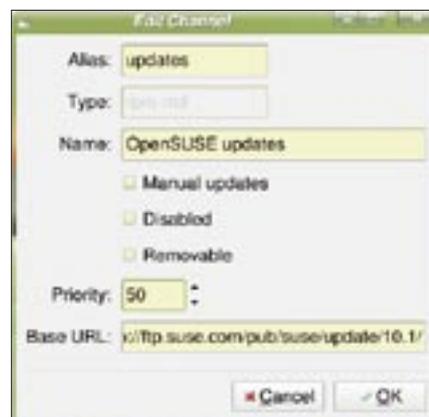


Figure 4: Creating a channel for Suse 10.1 updates.

## Tolerant Search

```
01 kim:~ # apt-get install
    realplay
02 Reading package list... done
03 Creating dependency tree...
    done
04 E: Couldn't find realplay
    package
05
06 kim:~ # smart install realplay
07 Loading cache...
08 Updating cache...
    ##### [100%]
09 ERROR!: 'realplay' does not
    match a package: suggest:
10 ERROR!:
    RealPlayer-10.0.7-10@i586
11 Saving cache...
```

*package* instead. If Smart starts to downgrade packages, you can assume that you have forgotten some priorities along the way. To remedy this, select *Edit | Priorities* in the GUI.

By default, Smart will delete downloaded package archives after installation. If you would prefer to keep the archives to reuse them, run the package manager at the command line, setting the *-o remove-packages = false* when you do. The downloaded packages will be stored in */var/lib/smart/packages/*, and you can use them to upgrade another machine. To generally prevent Smart from deleting downloads, just give the *smart config --set remove-packages = false* command.

If you are a KDE user, it might be worth your while to install the KSmart-tray kicker applet, which keeps you up to date with recent developments. If Smart does run into difficulty at some stage, there's an easy solution: *Edit | Fix All Problems...* will make Smart even smarter. ■

## INFO

- [1] Smart: <http://labix.org/smart>
- [2] Guru repository: <ftp://ftp.gwdg.de/pub/linux/misc/suser-guru/rpm/10.1/>
- [3] Guru RPM page: <http://linux01.gwdg.de/~pbleser/>
- [4] Converting apt lists to channel files: <http://linux01.gwdg.de/~pbleser/files/smart/apt-sources-to-smart.sh>