

Be a Zypper Power User

Seth Kenlon

There are “user-friendly” Linux distributions and there are “geek” distributions and there are “enterprise” distribution. Definitions of those vary from person to person, and there are probably more categories than even those three, but it’s no small feat that openSUSE manages to encompass all of these all in one clean, professional, and reliable package. One of the most important tools on a Linux distribution is its software management; how users find, install, update, and remove new software to their system. openSUSE’s *zypper* tool is one of the most flexible and most powerful interfaces of them all, and if you learn how to use it, then you’ll be able to manage your SUSE system like a pro.

Zypper is a command line software management tool; it is used to discover, install, update, and remove software from an openSUSE system. It’s hardly unique in that sense, and in fact it’s a latecomer by comparison. Rpm, apt-get, yum, emerge, and countless other shell-based package managers preceded zypper; some by decades. Fortunately, zypper’s developers leveraged

their belated arrival by learning from the tools that paved the way. Zypper consequently inherits most of the best aspects of all the other package management tools and leaves out all of the annoyances.

Yet Another System Tool

The default system configuration tool on openSUSE is YaST, a graphical and intuitive interface to nearly anything that can be configured on a Linux computer. Home users and systems administrators alike can appreciate the consolidated, centralized management that YaST offers.

YaST’s package manager isn’t exactly famous for its beautiful and creative interface. It is quite flexible, with many different options for searching criteria and with an abundance of information being available in many different tabs.

Recently, a cross-distribution technology called PackageKit has arisen, which offers a standardized interface for managing software. On KDE desktops, the name of the application is Apper, and openSUSE includes it

in case you prefer it to YaST. It’s a lot more appealing to look at, it makes searching easier, and the interface is generally less cluttered and more colourful.

However, power users might find YaST and Apper both inconvenient, since many of the functions in these interfaces must be done graphically. If you’re not running in a graphical environment, then that can be a problem; the last thing a power user wants to do is bring up a desktop sharing session just to install an application or perform an update. And even if a graphical environment is present, sometimes it’s just plain faster to type in exactly what the

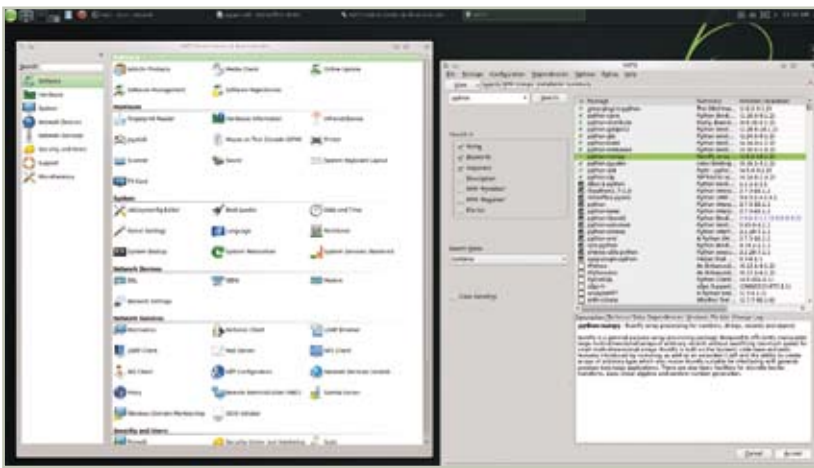


Figure 1. YaST isn’t the prettiest of package managers, but it is the default, and it’s very flexible

Script 1. Searching for a keyword within an application name only

```
$ zypper search paint
S | Name | Summary | Type
-----+-----+-----+-----
| kolourpaint | Paint Program | package
| kolourpaint | Paint Program | srcpackage
| mtpaint | Painting program for creating ico-> | package
| mtpaint-handbook | Handbook for the mtpaint painting-> | package
| mtpaint-lang | Languages for package mtpaint | package
| tuxpaint | Drawing Program for Young Children | package
| tuxpaint-config | Configuration tool for Tux Paint | package
| tuxpaint-config-lang | Languages for package tuxpaint-co-> | package
| tuxpaint-devel | Devel files of tuxpaint | package
```

computer should do than it is to click through arrays of icons and menus and windows and dialogue boxes.

The Libzypp Stack

Zypper is based on libzypp, the software library that manages packages on openSUSE. Libzypp can be used through YaST, Apper, or Zypper. In other words, all the different methods of installing software are simply different frontend interfaces to the same backend process.

It's been said that to most people, an "operating system" is invisible, and that what most people really mean when they talk about their OS is the collection of applications that they run on a daily basis. Zypper isn't an application that most people will use on a daily basis. It's more a tool that one will use all day for the first day that they spend configuring a system, and then will go weeks without using at all. Knowing how to use it, however, can make it a very useful tool, and understanding its many options makes it trivial to quickly install something that you suddenly find you unexpectedly need, and it will make it easier to discover new applications that you never knew existed.

Smarter Searches

To try out zypper, launch a terminal by click the openSUSE menu in the lower left of your KDE desktop. Select Konsole to run the default terminal.

The first thing you're likely to need is the ability to search for available packages. The syntax of a zypper command is generally: zypper + command + options + package.

That means that when you use zypper, you'll always type "zypper" first, and then some command like "search" or "install", and then any extra options you need to tell zypper, and finally the package you want those commands and options operated upon.

As an example, assume you are looking for a painting programme. You know about GIMP but you'd like

some alternatives. So you would type zypper search paint (Script 1).

As you can see, you receive quite a few results, all entirely relevant to your interests. However, zypper has only returned results with the term "paint" in the package name. To expand your search for even more results, tell zypper to search descriptions as well (notice also that you can abbreviate any zypper command

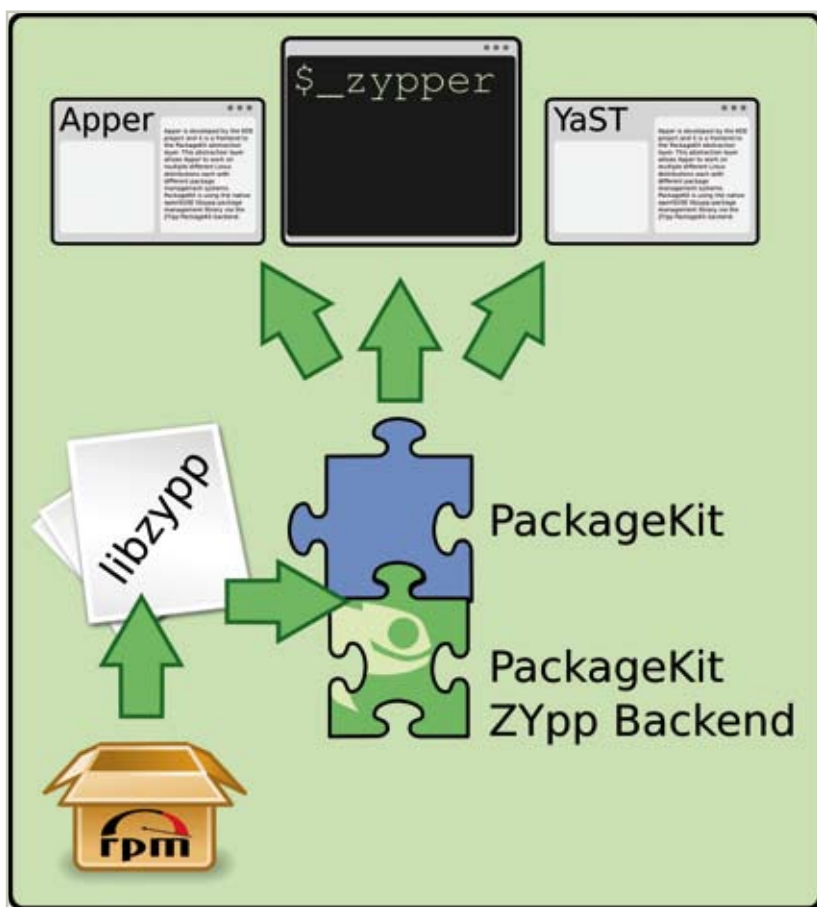


Figure 2. Package management is all based on libzypp in openSUSE, so it mostly doesn't matter what you use when you're running the show

Script 2. Searching for a keyword in the name and description

```
$ zypper se --search-descriptions paint
S | Name | Summary | Type
-----|-----|-----|-----
| blender | A 3D Modelling And Rendering Pack-> | package
| gcompris | J'ai compris / I Have Understood | package
| google-allerta-fonts | Easily Readable Sans Serif Font | package
| kdegraphics3-imaging | Image-Related KDE Applications | package
| khunphan | Traditional Thai Puzzle Board Gam-> | package
| kolourpaint | Paint Program | package
| kolourpaint | Paint Program | srcpackage
| mtpaint | Painting program for creating ico-> | package
| mtpaint-handbook | Handbook for the mtpaint painting-> | package
| mtpaint-lang | Languages for package mtpaint | package
i | netpbm | A Powerful Graphics Conversion Pa-> | package
| openSUSE-2013-477 | python-veusz: Two fixes | patch
| pinta | Simple Paint Application | package
| texlive-lecturer | On-screen presentations for (almo-> | package
| texlive-spot | Spotlight highlighting for Beamer | package
| tuxpaint | Drawing Program for Young Children | package
| tuxpaint-config | Configuration tool for Tux Paint | package
| tuxpaint-config-lang | Languages for package tuxpaint-co-> | package
| tuxpaint-devel | Devel files of tuxpaint | package
i | xf86-video-vooodoo | Voodoo video driver for the Xorg -> | package
i | xrefresh | Utility to refresh all or part of-> | package
```

to two-letter shortcuts; “search” becomes just “se”), see Script 2.

Clearly, searching descriptions reveals a few new gems that you’d not have known otherwise, such as Blender, Pinta, and netpbm.

Quicker Installs

Installing from zypper is simple, but does require root privileges, so wrap the install in an “su” command:

```
su -c 'zypper install pinta'
```

Zypper will request your root password, which is the administrative password you created at install time, and then it will process the order. Part of zypper’s processing is parsing the metadata of the RPM (a software installable package, like an .msi or some .exe’s on Windows or .pkg and .mpkg on Macs) to determine what

other software must be installed in order for your desired application to run correctly. On Windows and Mac, you may have encountered this, although not during install process; sometimes you’ll download an application and install it and then when you first launch it, the application informs you that you need to install Java or Flash or Silverlight or Ffmpeg or some other set of software in order for it to work. On Linux, this happens during install time instead.

Zypper will then pause, asking you if you really want to install the dependency software.

To speed things along, especially when a lot of software is being installed, you can add the “-y” (for “yes”) and “--auto-agree-with-licenses” to your install command so that zypper will blindly accept any and all questions:

```
su -c 'zypper in -y --auto-agree-with-licenses pinta'
```

After issuing that command, you can get up and walk away from your computer, secure in the knowledge that



Figure 3. Try finding a package called “pinta” with the search term “paint” in any other package manager. Conversely, try not finding it. You can do either, intentionally, with zypper!

Script 3. Searching a newly added, unofficial repository

```
$ zypper se ffmpeg
New repository or package signing key received:
Key ID: 45A1D0671ABD1AFB
Key Name: PackMan Project (signing key) <packman@links2linux.de>
Key Fingerprint: F8875B880D518B6B8C530D1345A1D0671ABD1AFB
Key Created: Mon 20 Sep 2010 04:37:32 PM EDT
Key Expires: Fri 19 Sep 2014 04:37:11 PM EDT
Repository: Packman
Do you want to reject the key, trust temporarily, or trust always? [r/t/a/?] (r):
```

Script 4. Can't remember what repositories you have configured? Zypper knows all!

```
$ zypper repos
# | Alias | Name | Enabled | Refresh
--+-+-----+-----+-----+-----
1 | Packman | Packman | Yes | No
2 | repo-debug | openSUSE-12.3-Debug | No | Yes
3 | repo-debug-update | openSUSE-12.3-Update-Debug | No | Yes
4 | repo-debug-update-non-oss | openSUSE-12.3-Update-Debug-Non-Oss | No | Yes
5 | repo-non-oss | openSUSE-12.3-Non-Oss | Yes | Yes
6 | repo-oss | openSUSE-12.3-Oss | Yes | Yes
7 | repo-source | openSUSE-12.3-Source | No | Yes
8 | repo-update | openSUSE-12.3-Update | Yes | Yes
9 | repo-update-non-oss | openSUSE-12.3-Update-Non-Oss | Yes | Yes
```

everything you listed will be installed. There's nothing more frustrating than starting a batch install of 100 packages to run overnight and then coming back in the morning only to find the entire job stalled after the second install because the installer is waiting for you to accept a license agreement for an obscure Microsoft font that you'll never even use anyway.

Repository Management

openSUSE, like all Linux distributions, is bound by the laws of the locality in which it is distributed. This means that within the borders of the US, it's a big legal gray area as to whether users are legally permitted to download things like audio and video decoders and encoders that were originally developed as proprietary code, even if those decoders and encoders have been reimplemented by third parties. Of course, with the internet being the internet, most users just add non-US package repositories to their list of available software and download the codecs from Europe or South America. Problem solved.

The question is how to get these non-US repositories onto your system.

It's a good question, and one that probably accounts for 33% of all internet traffic. Well, maybe not quite that much, but the number of tutorials on how to add repositories to yum or apt-get based distros is staggering, and it's a little frightening to see how many of those tutorials are out of date.

Zypper solves this issue by letting the user manage their repositories through zypper. By default, zypper is configured with the standard openSUSE servers, so when

you search for and install a program, zypper looks at those servers, searches through the applications stored there, and then downloads the package and installs it on your system. If you add more servers, then you can find more software.

One of the most popular unofficial repositories for openSUSE is called Packman. You can visit their site at <http://packman.links2linux.org> but of course this is Linux, literally the inventors of the "app store"

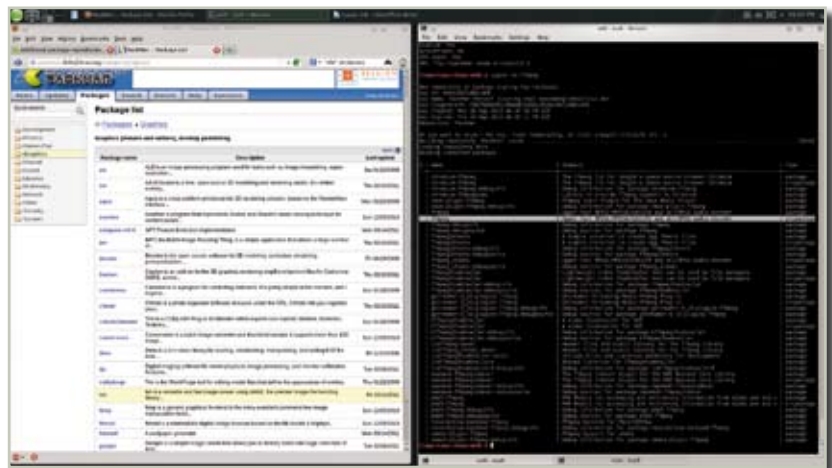


Figure 4. Packman offers plenty of software that openSUSE cannot. Add it now and see what you've been missing

concept, so you needn't download RPM files from the site and install them manually. Instead, use zypper to add the Packman repository.

The syntax for adding a repository is `zypper + addrepo` ("ar" for short) + URL of server + an alias for the repository.

To add the Packman repositories to your list of searchable servers:

```
su -c 'zypper ar ftp:// \
packman.inode.at/suse/12.3 \
Packman'
```

This command adds the Packman server to your list of available servers, and aliases it to the simpler term "Packman". To test it out, try searching for a package rife with codec debates, like `ffmpeg`, see Script 3.

The first time you search a repository, openSUSE will attempt to import its digital signing key. It's safe to accept it always [a], meaning that it won't bother you again until 19 September of 2014. If ever there is an unauthorized package on the server that has *not* been signed with that key, then zypper will be able to alert you, and you can decline to install the package until you are able to contact the developers over at Packman to find out why they're hosting a package that has not been digitally signed by them.

After you accept the key, a list of results will be found, and you'll discover just how much multimedia gold you've been missing out on by using only the standard SUSE servers.

You can also remove repositories. For instance, if you were trying out a new application from a company that maintained a repository just for their one application (Google Chrome, Adobe Flash, and a few other projects have been known to do this), you may decide later that not only do you not care to use the application but you also have no need for their repository.

For instance, if you had added a repository from Example, Inc. and aliased it to 'Example', and later decide you want to remove it, just use zypper's `removereпо` option:

```
$ su -c 'zypper rr Example'
```

If you're not sure what repositories you have configured, that too is a zypper command away (Script 4).

All the work you put into configuring your list of repositories should be backed up. Naturally, there's a zypper command for that:

```
$ zypper repos -e my.repo
```

Should you ever re-install or somehow lose your configuration, you can restore all of your repositories back into zypper with your backup file:

```
$ su -c 'zypper ar my.repo'
```

Fine-Grain Control

If you're particularly determined to install an available package from one specific repository, you can define which repository zypper should use with the `-r` flag, or `-repo` in long form:

```
$ su -c 'zypper install -r \
Packman xmms2-plugin-mp4'
```

And if you especially geeky and would prefer that your system actually compiles the software from its raw source code, then you can even tell zypper to do that with `source-install` ("si" for short):

```
$ su-c 'zypper si ffmpeg'
```

This, amazingly, pulls in all of the myriad dependencies for an `ffmpeg` build, installs them, and then compiles and installs `ffmpeg`. That's a half a gigabyte worth of software, some installed from binary and the main payload from source code, all in one command. If you're not in Geek Heaven from that, you should be.

The Zypper Shell

While zypper is probably most useful because of its non-interactive ability, sometimes you just want to sit down and spend some serious quality time with your system. If you're configuring a system with new software and lists of repositories, then eventually you'll get tired of constantly typing in 'zypper' at the beginning of every command.

For this, there is the zypper shell. In the zypper shell, you can issue commands directly to zypper without constantly preceding it with 'zypper' and 'su -c' and so on.

```
$ su -c 'zypper shell'
zypper>
```

But Wait, There's More

This article has barely touched on the full capabilities of the zypper command. With zypper, you can manage patch releases, pre-defined patterns for logically associated packages, package logs to prevent an especially important software from being updated, download source code, and even upgrade your entire SUSE distribution.

Zypper is one of the best command line package managers available. Try it today, and start learning all of its many options. ■



Figure 5. If screenshots like this get you excited, you may be a geek. Also, you'll love zypper