

Zypper tips

To check from which repo package came from

```
# rpm -qa --qf '%-20{NAME} %-20{VENDOR}\n' 'libav*5[678]'
```

To install only packages from one repo

```
#zypper in -r pvdm-kde43 amarok
```

Change the behaviour of download-install-download-install-download to download all-install all

modify /etc/zypp/zypp.conf and find and change:

```
commit.downloadMode = DownloadInAdvance
```

- [How to locally cache zypper update files](#)

upgrading your system using 'zypper dup'

- When you want to do an upgrade, "zypper dup" actually brings your system to the version of the available packages. This may mean downgrading and removing of installed packages.
- The recommended way of doing this is to perform it against a specific repo: use

```
zypper dup -from repo -r, --repo <alias|#|URI>    Load only the specified repository.  
zypper dup -r 5 -r 6 -r 7
```

- If you intend to use "zypper dup" without any specific repository, you need to define repo priorities if you don't want to encounter surprises.
- For the full story read <http://duncan.mac-vicar.com/blog/archives/619> here.

=if you want to find a certain [file:=](#)

```
zypper in 'libqtiff.so()(64bit)'  
Reading installed packages...  
'libqt4-x11' providing 'libqtiff.so()(64bit)' is already installed.  
Nothing to do.
```

if you want to install a certain version:

```
zypper in MozillaFirefox \>= 3
```

Other examples:

```
zypper in yast*                # install all yast modules  
zypper in -  
t pattern lamp_server        # install lamp_server pattern (packages needed for a LAMP server)  
zypper in vim -emacs         # install vim and remove emacs in one go  
zypper in amarok packman:libxine1 # install libxine1 from packman and amarok from any repo  
zypper in bitchx-1.1-81.x86_64.rpm # install bitchx rpm from local directory  
zypper in -f subversion      # force reinstallation of subversion
```



```
zypper info -t patch MozillaFirefox # show information about 'MozillaFirefox' patch
zypper patch-info MozillaFirefox # the same as above
zypper info -t pattern lamp_server # show info about 'lamp_server' pattern
zypper info -t product openSUSE-FTP # show info about specified product
```

dependencies

what-provides or wp

To list all providers of specified capability, do:

```
zypper wp firefox
Reading installed packages...
S | Name          | Type   | Version   | Arch  | Repository
--+-+-----+-----+-----+-----+-----
i | MozillaFirefox | package | 3.0-0.1   | x86_64 | Updates for 11.1
v | MozillaFirefox | package | 2.9.95-25.1 | x86_64 | openSUSE-11.1-0ss
v | MozillaFirefox | package | 3.0-0.1   | i586  | Updates for 11.1
v | MozillaFirefox | package | 2.9.95-25.1 | i586  | openSUSE-11.1-0ss
```

Regarding optional dependencies, we do have them, and use them. They're enabled by default. To avoid the several hundred unexpected packages like in the example in the article you can use zypper with `--no-recommends` or turn them off in the `zypper.conf` config file.

The zypper summary before an installation distinguishes between hard dependencies and recommends, so you can see what turning recommends off would have saved you even if you didn't.

package locks

To lock all packages starting with 'yast2', do:

```
zypper al 'yast2*'
Reading installed packages...
Specified lock has been successfully added
```

To list currently active locks:

```
zypper ll
# | Name          | Type   | Repository
--+-+-----+-----+-----
1 | libpoppler3   | package | (any)
```

```
2 | libpoppler-glib3 | package | (any)
```

```
3 | yast* | package | (any)
```

To remove a lock, do:

```
zypper rl yast2-packager
```

Other examples:

```
zypper al zypper # lock package 'zypper' (exact match)
```

```
zypper al -r repo-oss virtualbox* # restrict the lock to 'repo-
```

```
oss' repository (allowing installation from others)
```

```
zypper rl 3 # remove lock by number
```

rpm examples

```
# rpm -qi amarok Show details of package
```

```
# rpm -qf /usr/sbin/sshd Shows where file came from
```

```
# rpm -qR openssh-server Show dependencies - What do I need for this package to work?
```

```
# rpm --whatrequires initscripts Find reverse dependencies -
```

```
What will break if I remove this package
```

```
# rpm -qip Show details of not yet installed package
```

```
# rpm --verify openssh-
```

```
server Find out if any files in the package have been modified since install.
```

```
# rpm -qlv postfix list all files in a rpm
```

Revision #3

Created 2026-04-01 17:11:41 CEST by Philip

Updated 2026-04-13 19:25:46 CEST by Philip