

Zypper Cheat Sheet

More Information:

<http://en.opensuse.org/Zypper/Usage> or type `man zypper` on a terminal

Page 1



For Zypper version 1.0.9

Basic Help

`zypper` #list the available global options and commands
`zypper help [command]` #Print help for a specific command
`zypper shell` or `zypper sh` #Open a zypper shell session

Repository Management

Listing Defined Repositories

`zypper repos` or `zypper lr`

Examples:

`zypper lr -u` #include repo URI on the table
`zypper lr -P` #include repo priority and sort by it

Refreshing Repositories

`zypper refresh` or `zypper ref`

Examples:

`zypper ref packman main` #specify repos to be updated
`zypper ref -f upd` #force update of repo 'upd'

Modifying Repositories

`zypper modifyrepo` or `zypper mr`

Examples:

`zypper mr -d 6` #disable repo #6
`zypper mr -rk -p 70 upd` #enable autorefresh and rpm files 'caching' for 'upd' repo and set its priority to 70
`zypper mr -Ka` #disable rpm files caching for all repos
`zypper mr -kt` #enable rpm files caching for remote repos

Adding Repositories

`zypper addrepo` or `zypper ar` #followed by the repo url and alias

Example:

`zypper ar http://download.opensuse.org/update/11.1/ update`

Removing Repositories

`zypper removerepo` or `zypper rr`

Examples:

`zypper rr packman main`

Renaming Repositories (for the alias only)

`zypper namerepo` or `zypper nr`

Examples:

`zypper nr 3 upd`

Package Management

Selecting Packages

By capability name:

`zypper in 'perl(Log::Log4perl)'`

`zypper in qt`

By capability name and/or architecture and/or version

`zypper in 'zypper<0.12.10'`

`zypper in zypper.i586=0.12.11`

By exact package name (--name)

`zypper in -n ftp`

By exact package name and repository (implies --name)

`zypper in factory:zypper`

By package name using wildcards

`zypper in yast*ftp*`

By specifying a .rpm file to install

`zypper in skype-2.0.0.72-suse.i586.rpm`

Installing Packages

`zypper install` or `zypper in`

Examples:

`zypper install git`

By capability they provide

`zypper in MozillaFirefox < 3`

Others

`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

`zypper in amarok upd:libxine1` #install libxine1 from upd

Removing Packages

`zypper remove` or `zypper rm`

Examples:

`zypper remove sqlite`

Export/Import Repositories

`zypper repos --export` or `zypper lr -e`

Examples:

`zypper lr --export backups/repos/foo.repo`

`zypper ar backups/repos/foo.repo` #import

Source Packages and Build Dependencies

`zypper source-install` or `zypper si`

Examples:

`zypper si zypper`

Install only the source package

`zypper in -D zypper`

Install only the build dependencies

`zypper in -d zypper`

Updating Packages

`zypper update` or `zypper up`

Examples:

`zypper up` #update all installed packages with newer version as far as possible

`zypper up libzypp zypper` #update libzypp and zypper

`zypper in sqlite3` #update sqlite3 or install if not yet installed

Zypper in Scripts and Applications

Non Interactive Mode

`zypper --non-interactive`

Examples:

`zypper --non-interactive patch` #skips all interactive patches which would require user confirmation

No GPG Checks Mode

`zypper --no-gpg-checks`

Auto-agree with Licenses

`zypper --auto-agree-with-licenses`

Quiet Output

`zypper --quiet`

XML Output

`zypper --zmlout`

Zypper Cheat Sheet

More Information:

<http://en.opensuse.org/Zypper/Usage> or type `man zypper` on a terminal

Page 2



For Zypper version 1.0.9

Querying

Searching Packages

`zypper search` or `zypper se`

Examples:

`zypper se -dC --match-words RSI` #look for RSI acronym (case-sensitively), also in summaries and descriptions
`zypper se 'yast*'` #show all packages starting with 'yast'
`zypper se -r upd` #list all packages from 'upd' repository
`zypper se -i sqlite` #show all 'sqlite' installed packages
`zypper se -t pattern -r upd` #list all patterns available in the 'upd' repository

Getting Information about Packages

`zypper info` or `zypper if`

Examples:

`zypper info amarok`
`zypper info -t patch amarok` #show info for 'amarok' patch
`zypper patch-info amarok` #same as above
`zypper info -t pattern lamp_server` #info 'lamp_server' pattern

Getting Information about Dependencies

`zypper what-provides` or `zypper wp`

Examples:

`zypper wp firefox`

Utilities

Verify Dependencies

`zypper verify` or `zypper ve`

Note:

This is useful in cases of a broken system

Install New Recommended Packages

`zypper install-new-recommends` or `zypper inr`

Package Locks

Lock Packages

`zypper addlock` or `zypper al`

Examples:

`zypper al 'yast2*'` #lock all packages starting with 'yast2'

Remove Locks

`zypper removelock` or `zypper rl`

Examples:

`zypper rl 'yast2*'` #remove locks to all packages starting with 'yast2'

List Locks

`zypper locks` or `zypper ll`

Update Management

Listing Needed Patches

`zypper list-patches` or `zypper lp`

Applying Patches

`zypper patch`

Listing All Patches

`zypper patches`

Checking Patches

`zypper patch-check` or `zypper pchk`

Getting Information About Patches

`zypper patch-info`

`zypper info -t patch`

Packages Updates

`zypper list-updates` or `zypper lu`

`zypper update` or `zypper up`

Distribution Upgrade

`zypper dist-upgrade` or `zypper dup`

Note:

When doing a distribution update, the best is to work only with the repositories of the distribution you want to install.

Vocabulary

Repositories

HTTP or FTP server, DVD, or a folder on a local disc. where a group or set of packages are located.

Resource Identifiers (URI)

To specify locations of repositories or other resources (RPM files, .repo files) you can use any type of URIs supported by libzypp. See <http://en.opensuse.org/Libzypp/URI> for a complete list and usage examples.

Refresh

Refreshing a repository means downloading metadata of packages from the medium (if needed), storing it in local cache (typically under `/var/cache/zypp/raw/<alias>` directory) and preparing the metadata into .solvable files (building the solvable cache), typically under `/var/cache/zypp/solvable/<alias>`.

Services

Services are one level above repositories and serve to manage repositories or to do some special tasks. Libzypp currently supports only one type of services, the Repository Index Service (RIS).

Package Types

zypper works with several types of resource objects, called resolvables. A resolvable is a package, patch, pattern, or a product.

package - an ordinary RPM package

patch - update of one or more packages.

pattern - group of packages required or recommended to install some functionality

product - group of packages which are necessary to install a product