suse, tumbleweed, kernel, versions, pin, keep, multiversionkernel, retention

Keeping Multiple Kernel Versions On openSUSE Tumbleweed

- https://en.opensuse.org/SDB:Keep_multiple_kernel_versions
- https://lizards.opensuse.org/2011/07/14/improved-kernel-package-retention-in-12-1/

Updating openSUSE Tumbleweed will also install new kernel versions. Sometimes this will break things. To keep my system running I use the "**multiversion kernel**" feature of openSUSE which is available since version 12.1.

This feature makes it possible to

- delete an old kernel only after the system has been rebooted successfully with the new kernel
- keep a specified number of older kernels as spares
- keep a specific kernel version

List Installed Kernels

To know the exact kernel version (in case you want to pin it):

```
zypper se -s kernel-default
```

```
Loading repository data...
Reading installed packages...

S  | Name                          | Type    | Version          | Arch   | Repository
---+-------------------------------+---------+------------------+--------+----------------------
i+ | kernel-default                | package | 5.14.11-2.1      | x86_64 | (System Packages)
i+ | kernel-default                | package | 5.14.14-1.1      | x86_64 | Main Repository (OSS)
i+ | kernel-default                | package | 5.14.14-1.1      | x86_64 | openSUSE:Factory
i+ | kernel-default                | package | 5.14.14-1.1      | x86_64 | openSUSE:Tumbleweed
i+ | kernel-default                | package | 5.14.14-1.1      | x86_64 | openSUSE-20210527-0
v  | kernel-default                | package | 5.14.14-1.1      | i586   | Main Repository (OSS)
v  | kernel-default                | package | 5.14.14-1.1      | i586   | openSUSE:Factory
v  | kernel-default                | package | 5.14.14-1.1      | i586   | openSUSE:Tumbleweed
v  | kernel-default                | package | 5.14.14-1.1      | i586   | openSUSE-20210527-0
   | kernel-default-base           | package | 5.14.14-1.1.22.7 | x86_64 | Main Repository (OSS)
   | kernel-default-base           | package | 5.14.14-1.1.22.7 | x86_64 | openSUSE:Factory
   | kernel-default-base           | package | 5.14.14-1.1.22.7 | x86_64 | openSUSE:Tumbleweed
   | kernel-default-base           | package | 5.14.14-1.1.22.7 | x86_64 | openSUSE-20210527-0
   | kernel-default-base           | package | 5.14.14-1.1.22.7 | i586   | Main Repository (OSS)
   | kernel-default-base           | package | 5.14.14-1.1.22.7 | i586   | openSUSE:Factory
   | kernel-default-base           | package | 5.14.14-1.1.22.7 | i586   | openSUSE:Tumbleweed
   | kernel-default-base           | package | 5.14.14-1.1.22.7 | i586   | openSUSE-20210527-0
   | kernel-default-base-rebuild   | package | 5.14.14-1.1.22.7 | x86_64 | Main Repository (OSS)
   | kernel-default-base-rebuild   | package | 5.14.14-1.1.22.7 | x86_64 | openSUSE:Factory
   | kernel-default-base-rebuild   | package | 5.14.14-1.1.22.7 | x86_64 | openSUSE:Tumbleweed
   | kernel-default-base-rebuild   | package | 5.14.14-1.1.22.7 | x86_64 | openSUSE-20210527-0
   | kernel-default-base-rebuild   | package | 5.14.14-1.1.22.7 | i586   | Main Repository (OSS)
   | kernel-default-base-rebuild   | package | 5.14.14-1.1.22.7 | i586   | openSUSE:Factory
   | kernel-default-base-rebuild   | package | 5.14.14-1.1.22.7 | i586   | openSUSE:Tumbleweed
   | kernel-default-base-rebuild   | package | 5.14.14-1.1.22.7 | i586   | openSUSE-20210527-0
   | kernel-default-devel          | package | 5.14.14-1.1      | x86_64 | Main Repository (OSS)
   | kernel-default-devel          | package | 5.14.14-1.1      | x86_64 | openSUSE:Factory
   | kernel-default-devel          | package | 5.14.14-1.1      | x86_64 | openSUSE:Tumbleweed
   | kernel-default-devel          | package | 5.14.14-1.1      | x86_64 | openSUSE-20210527-0
   | kernel-default-devel          | package | 5.14.14-1.1      | i586   | Main Repository (OSS)
   | kernel-default-devel          | package | 5.14.14-1.1      | i586   | openSUSE:Factory
   | kernel-default-devel          | package | 5.14.14-1.1      | i586   | openSUSE:Tumbleweed
   | kernel-default-devel          | package | 5.14.14-1.1      | i586   | openSUSE-20210527-0
```

Enabling the multiversion kernel feature

```
vim /etc/zypp/zypp.conf
```

see lines 554 and 555

```
## Configuration file for software management
## /etc/zypp/zypp.conf
##
## Boolean values are 0 1 yes no on off true false


[main]


##
## Override the detected architecture
##
## Valid values:  i586, i686, x86_64, ppc, ppc64, ia64, s390, s390x, ..
## Default value: Autodetected
##
## ** CAUTION: Only set if you know what you're doing !
## ** Changing this needs a full refresh (incl. download)
## ** of all repository data.
##
# arch = s390


##
## Path where the caches are kept.
##
## Valid values: A directory
## Default value: /var/cache/zypp
##
# cachedir = /var/cache/zypp


##
## Path where the repo metadata is downloaded and kept.
##
## Valid values: A directory
## Default value: {cachedir}/raw
##
## Changing this needs a full refresh (incl. download) of all repository data
##
# metadatadir = /var/cache/zypp/raw


##
## Path where the repo solv files are created and kept.
##
## Valid values: A directory
## Default value: {cachedir}/solv
##
# solvfilesdir = /var/cache/zypp/solv


##
## Path where the repo packages are downloaded and kept.
##
## Valid values: A directory
## Default value: {cachedir}/packages
##
# packagesdir = /var/cache/zypp/packages
```

```
##
## Path where the configuration files are kept.
##
## Valid values: A directory
## Default value: /etc/zypp
##
# configdir = /etc/zypp

##
## Path where the known repositories .repo files are kept
##
## Valid values: A directory
## Default value: {configdir}/repos.d
##
## Changing this invalidates all known repositories
##
# reposdir = /etc/zypp/repos.d

##
## Path where the known services .service files are kept
##
## Valid values: A directory
## Default value: {configdir}/services.d
##
## Changing this invalidates all known services
##
# servicesdir = /etc/zypp/services.d

##
## Path where custom repo variable definitions are kept
##
## Valid values: A directory
## Default value: {configdir}/vars.d
##
## Changing this undefines all custom repo variables. Built-in
## variables (like '$arch', '$basearch' or $releasever) are not
## affected, but reset to their default values.
##
## A custom repo variable is defined by creating a file inside the
## directory. The variable name equals the file name. The files fist
## line (up to but not including the newline character) defines the
## variables value.
##
# varsdir = /etc/zypp/vars.d

##
## Whether repository urls should be probed when added
##
## Valid values: boolean
## Default value: false
##
## If true, accessability of repositories is checked immediately (when added)
##   (e.g. 'zypper ar' will check immediately)
## If false, accessability of repositories is checked when refreshed
##   (e.g. 'zypper ar' will delay the check until the next refresh)
##
# repo.add.probe = false


##
## Amount of time in minutes that must pass before another refresh.
##
## Valid values: Integer
## Default value: 10
##
## If you have autorefresh enabled for a repository, it is checked for
## up-to-date metadata not more often than every <repo.refresh.delay>
## minutes. If an automatic request for refresh comes before <repo.refresh.delay>
## minutes passed since the last check, the request is ignored.
##
```

```
## A value of 0 means the repository will always be checked. To get the opposite
## effect, disable autorefresh for your repositories.
##
## This option has no effect for repositories with autorefresh disabled, nor for
## user-requested refresh.
##
# repo.refresh.delay = 10

##
## Translated package descriptions to download from repos.
##
## A list of locales for which translated package descriptions should
## be downloaded, in case they are availavble and the repo supports this.
## Not all repo formats support downloading specific translations only.
##
## Valid values:  List of locales like 'en', 'en_US'...
## Default value: RequestedLocales
##
## If data for a specific locale are not available, we try to find some
## fallback. Translations for 'en' are always downloaded.
##
# repo.refresh.locales = en, de

##
## Maximum number of concurrent connections to use per transfer
##
## Valid values: Integer
## Default value: 5
##
## This setting is only used if more than one is possible
## Setting it to a reasonable number avoids flooding servers
##
# download.max_concurrent_connections = 5

##
## Sets the minimum download speed (bytes per second)
## until the connection is dropped
## This can be useful to prevent security attacks on hosts by
## providing updates at very low speeds.
##
## 0 means no limit
##
# download.min_download_speed = 0

## Maximum download speed (bytes per second)
## 0 means no limit
# download.max_download_speed = 0

## Number of tries per download which will be
## done without user interaction
## 0 means no limit (use with caution)
# download.max_silent_tries = 5

##
## Maximum time in seconds that you allow a transfer operation to take.
##
## This is useful for preventing your batch jobs from hanging for hours due
## to slow networks or links going down. Limiting operations to less than a
## few minutes risk aborting perfectly normal operations.
##
## Valid values:  [0,3600]
## Default value: 180
##
# download.transfer_timeout = 180

##
## Whether to consider using a .delta.rpm when downloading a package
##
## Valid values: boolean
```

```
## Default value: true
##
## Using a delta rpm will decrease the download size for package updates
## since it does not contain all files of the package but only the binary
## diff of changed ones. Recreating the rpm package on the local machine
## is an expensive operation (memory,CPU). If your network connection is
## not too slow, you benefit from disabling .delta.rpm.
##
# download.use_deltarpm = true

##
## Whether to consider using a deltarpm even when rpm is local
##
## Valid values: boolean
## Default value: false
##
## This option has no effect unless download.use_deltarpm is set true.
##
#  download.use_deltarpm.always = false

##
## Hint which media to prefer when installing packages (download vs. CD).
##
## Valid values:        download, volatile
## Default value:       download
##
## Note that this just a hint. First of all the solver will choose the 'best'
## package according to its repos priority, version and architecture. But if
## there is a choice, we will prefer packages from the desired media.
##
## Packages available locally are always preferred. The question is whether
## you prefer packages being downloaded via FTP/HTTP/HTTPS (download), rather
## than being prompted to insert a CD/DVD (volatile), in case they are available
## on both media.
##
##   Name             | Priority | URI
##   openSUSE-11.1      99         dvd:///
##   openSUSE-11.1-Oss  99         http://download.opensuse.org/distribution/11.1/repo/oss
##
## In the above example 2 repositories with similar content are used. Rather
## than raising the priority of one of them to 'prefer' a certain media, you
## should use the same priority for both and set download.media_preference
## instead.
##
## download.media_preference = download

##
## Path where media are preferably mounted or downloaded
##
## Valid values:        A (writable) directory
## Default value:       /var/adm/mount
##
## The media backend will try to organize media mount points and download areas
## below this directory, unless a different location is requested by the application.
##
## If the directory is not accessible and read/writable for a specific user,
## the fallback is to use /var/tmp.
##
## download.media_mountdir = /var/adm/mount

##
## Signature checking (repo metadata and downloaded rpm packages)
##
##   boolean    gpgcheck        (default: on)
##   boolean    repo_gpgcheck   (default: unset -> according to gpgcheck)
##   boolean    pkg_gpgcheck    (default: unset -> according to gpgcheck)
##
## Explicitly setting 'gpgcheck', 'repo_gpgcheck' 'pkg_gpgcheck' in a
## repositories .repo file will overwrite the defaults for this specific
## repo.
```

```
##
## If 'gpgcheck' is 'on' (the default) we will check the signature of repo metadata
## (packages are secured via checksum inside the metadata). Using unsigned repos
## needs to be confirmed.
## Packages from signed repos are accepted if their checksum matches the checksum
## stated in the repo metadata.
## Packages from unsigned repos need a valid gpg signature, using unsigned packages
## needs to be confirmed.
##
## The above default behavior can be tuned by explicitly setting 'repo_gpgcheck'
## and/or 'pkg_gpgcheck':
##
##   'repo_gpgcheck = on' same as the default.
##
##   'repo_gpgcheck = off' will silently accept unsigned repos. It will NOT turn off
##   signature checking on the whole, nevertheless it's not a secure setting.
##
##   'pkg_gpgcheck = on' will enforce the package signature checking and the need
##   to confirm unsigned packages for all repos (signed and unsigned).
##
##   'pkg_gpgcheck = off' will silently accept unsigned packages. It will NOT turn off
##   signature checking on the whole, nevertheless it's not a secure setting.
##
## If 'gpgCheck' is 'off' (not recommended), no checks are performed. You can still
## enable them individually by setting 'repo_gpgcheck' and/or 'pkg_gpgcheck' to 'on'.
##
##   DISABLING GPG CHECKS IS NOT RECOMMENDED.
##   Signing data enables the recipient to verify that no modifications
##   occurred after the data were signed. Accepting data with no, wrong
##   or unknown signature can lead to a corrupted system and in extreme
##   cases even to a system compromise.
##
# repo_gpgcheck = unset -> according to gpgcheck
# pkg_gpgcheck =  unset -> according to gpgcheck

##
## Commit download policy to use as default.
##
##  DownloadOnly,       Just download all packages to the local cache.
##                      Do not install. Implies a dry-run.
##
##  DownloadInAdvance,  First download all packages to the local cache.
##                      Then start to install.
##
##  DownloadInHeaps,    Similar to DownloadInAdvance, but try to split
##                      the transaction into heaps, where at the end of
##                      each heap a consistent system state is reached.
##
##  DownloadAsNeeded    Alternating download and install. Packages are
##                      cached just to avid CD/DVD hopping. This is the
##                      traditional behaviour.
##
##  <UNSET>             If a value is not set, empty or unknown, we pick
##                      some sane default.
##
## commit.downloadMode =

##
## Defining directory which contains vendor description files.
##
## Each file in this directory defines a (comma separated) list of
## equivalent vendors string prefixes (case-insensitive comparison):
## ------------------------- file begin ----------------------
## [main]
## vendors = MyVendor,AlternateName
## ------------------------- file end ----------------------
## By this vendor strings starting with "MyVendor" or "AlternateName"
## are considered to be equivalent. Packages from equivalent vendors
## may replace each other without being considered as a 'vendor change'.
```

```
##
## NOTE: Within the "opensuse*" namespace exact matches (case insensitive)
## are required. "vendors = suse,opensuse" will allow switching between
## "suse*" and "opensuse", but not e.g. "opensuse build service".
##
## Valid values: A directory
## Default value: {configdir}/vendors.d
##
# vendordir = /etc/zypp/vendors.d


##
## The solvers general attitude when resolving jobs.
##
## Valid values:
##
##   Job       - Focus on installing the best version of the requested packages.
##               Add missing dependencies as needed. This is the solvers default.
##
##   Installed - Focus on applying as little changes to the installed packages
##               as needed. Choosing an older version of a package is valid if
##               it's dependencies require less changes to the system.
##
##   Update    - Focus on updating requested packages and their dependencies as
##               much as possible.
##
##   Default   - If the value is not set, empty or unknown, we use the default.
##
# solver.focus =

##
## Whether only required packages are installed.
##
## Recommended packages, will not be regarded.
##
## Valid values: boolean
## Default value: false
##
# solver.onlyRequires = false

##
## EXPERTS ONLY: Per default the solver will not replace packages of
## different vendors, unless you explicitly ask to do so. Setting this
## option to TRUE will disable this vendor check (unless the application
## explicitly re-enables it). Packages will then be considered based on
## repository priority and version only. This may easily damage your system.
##
## Valid values:  boolean
## Default value: false
##
# solver.allowVendorChange = false

##
## EXPERTS ONLY: TUNE DISTRIBUTION UPGRADE (DUP)
## Set whether to allow package version downgrades upon DUP.
##
## Valid values:  boolean
## Default value: true
##
# solver.dupAllowDowngrade = true

##
## EXPERTS ONLY: TUNE DISTRIBUTION UPGRADE (DUP)
## Set whether follow package renames upon DUP.
##
## Valid values:  boolean
## Default value: true
##
# solver.dupAllowNameChange = true
```

```
##
## EXPERTS ONLY: TUNE DISTRIBUTION UPGRADE (DUP)
## Set whether to allow changing the packages architecture upon DUP.
##
## Valid values:  boolean
## Default value: true
##
# solver.dupAllowArchChange = true

##
## EXPERTS ONLY: TUNE DISTRIBUTION UPGRADE (DUP)
## Set whether to allow changing the packages vendor upon DUP. If you
## are following a continuous distribution like Tumbleweed or Factory
## where you use 'zypper dup --no-allow-vendor-change' quite frequently,
## you may indeed benefit from disabling the VendorChange. Packages from
## OBS repos will then be kept rather than being overwritten by Tumbleweeds
## version.
##
## Valid values:  boolean
## Default value: true
##
solver.dupAllowVendorChange = false

##
## EXPERTS ONLY: Cleanup when deleting packages. Whether the solver should
## per default try to remove packages exclusively required by the ones he's
## asked to delete.
##
## This option should be used on a case by case basis, enabled via
## command line options or switches the applications offer. Changing
## the global default on a system where unattended actions are performed,
## may easily damage your system.
##
## CHANGING THE DEFAULT IS NOT RECOMMENDED.
##
## Valid values:  boolean
## Default value: false
##
# solver.cleandepsOnRemove = false

##
## This file contains requirements/conflicts which fulfill the
## needs of a running system.
## For example the system would be broken if not glibc or kernel is
## installed.
## So the user will be informed if these packages will be deleted.
##
## Format: Each line represents one dependency:
##         e.g.
##         requires:kernel
##         requires:glibc
## Default value: {configdir}/systemCheck
##
# solver.checkSystemFile = /etc/zypp/systemCheck

##
## This directory can contain files that contain requirements/conflicts
## which fulfill the needs of a running system (see checkSystemFile).
##
## Files are read in alphabetical order.
##
## Default value: {configdir}/systemCheck.d
##
# solver.checkSystemFileDir = /etc/zypp/systemCheck.d

##
## When committing a dist upgrade (e.g. 'zypper dup') a solver testcase
## is written to /var/log/updateTestcase-<date>. It is needed in bugreports.
## This option returns the number of testcases to keep on the system. Old
```

```
## cases will be deleted, as new ones are created.
##
## Use 0 to write no testcase at all, or -1 to keep all testcases.
##
## Valid values:        Integer
## Default value:       2
##
# solver.upgradeTestcasesToKeep = 2

##
## Whether dist upgrade should remove a products dropped packages.
##
## A new product may suggest a list of old and no longer supported
## packages (dropped packages). Performing a dist upgrade the solver
## may try to delete them, even if they do not cause any dependency
## problem.
##
## Turning this option off, the solver will not try to remove those
## packages unless they actually do cause dependency trouble. You may
## do the cleanup manually, or simply leave them installed as long
## as you don't need the disk space.
##
## Valid values:        Boolean
## Default value:       true
##
# solver.upgradeRemoveDroppedPackages = true

##
## Packages which can be installed in different versions at the same time.
##
## Packages are selected either by name, or by provides. In the later case
## the string must start with "provides:" immediately followed by the capability.
##
## Example:
##      kernel                          - just packages whith name 'kernel'
##      provides:multiversion(kernel)   - all packages providing 'multiversion(kernel)'
##                                        (kenel and kmp packages should do this)
## Valid values:
##      Comma separated list of packages.
##
## Default value:
##      empty
##
multiversion = provides:multiversion(kernel)

##
## Defining directory which may contain additional multiversion definitions.
##
## If the directory exists, each file in this directory is scanned, expecting
## one valid multiversion list entry per line. Empty lines and lines starting
## with '#' are ignored.
## ------------------------- [/etc/zypp/multiversion.d/example file begin] ----------------------
## # An alternate way to enable kernel packages being
## # installed in parallel:
##
## provides:multiversion(kernel)
## ------------------------- [/etc/zypp/multiversion.d/example file end] ----------------------
##
## Valid values: A directory
## Default value: {configdir}/multiversion.d
##
# multiversiondir = /etc/zypp/multiversion.d

## Comma separated list of kernel packages to keep installed in parallel, if the
## above multiversion variable is set. Packages can be specified as
## 2.6.32.12-0.7 - Exact version to keep
## latest        - Keep kernel with the highest version number
## latest-N      - Keep kernel with the Nth highest version number
## running       - Keep the running kernel
## oldest        - Keep kernel with the lowest version number (the GA kernel)
```

```
## oldest+N      - Keep kernel with the Nth lowest version number
##
## Note: This entry is not evaluated by libzypp, but by the
##       purge-kernels service (via /sbin/purge-kernels).
##
## Default: Do not delete any kernels if multiversion = provides:multiversion(kernel) is set
# multiversion.kernels = latest,latest-1,running
multiversion.kernels = latest,latest-1,latest-2,running,5.14.11-2.1

##
## Path to locks file. If not exist then is create.
## In this file is saved also UI locks.
##
## valid value: path to file or place where file can be created
## default value: {configdir}/locks
##
# locksfile.path = /etc/zypp/locks

##
## Whether to apply locks in locks file after zypp start.
##
## Valid values: boolean
## Default value: true
##
# locksfile.apply = true

##
## Where update items are stored
## (example: scripts, messages)
##
## Valid values: path to directory
## Default value: /var/adm
##
# update.datadir = /var/adm

##
## Where update messages are stored
##
## Valid values: path to directory
## Default value: {update.datadir}/update-messages
##
# update.messagesdir = /var/adm/update-messages

##
## Where update scripts are stored
##
## Valid values: path to directory
## Default value: {update.datadir}/update-scripts
##
# update.scriptsdir = /var/adm/update-scripts

##
## Command to be invoked to send update messages.
##
## Packages may leave an update message file in {update.messagesdir}.
## At the end of each commit, zypp collects those messages and may send
## a notification to the user.
##
## zypp will prepare the update messages according to the selected
## content format and pipe the content to the command.
##
## Format:
##     single - For each update message invoke the command and send
##              the message.
##     none   - For each update message invoke the command but don't
##              use a pipe to send any data. You probably want to pass
##              the message file on the commandline using %P (see
##              Substitutions).
##     digest - Single invocation of the command, sending the path
```

```
##              names of all update message. One per line.
##     bulk   - Single invocation of the command, sending the
##              concatenated content of all update messages, separated
##              by Ctrl-L.
##
## Substitutions:
##     %p     - package identification (name-version-release.arch)
##     %P     - full path to the update message file
##
## Valid values: The value is specified as "format | command".
##               An empty value will turn off any notification.
##
## Examples:     single | mail -s 'Update message from %p' root
##               none   | my-send-script -f %P
##
## Default value: <empty>
##
# update.messages.notify =

##
## Options for package installation: excludedocs
##
## Don't install any files which are marked as documentation.
##
## Valid values:  boolean
## Default value: no
##
# rpm.install.excludedocs = no

##
## Location of history log file.
##
## The history log is described at
## http://en.opensuse.org/Libzypp/Package_History
##
## Valid values: absolute path to a file
## Default value: /var/log/zypp/history
##
# history.logfile = /var/log/zypp/history

##
## Global credentials directory path.
##
## If a URL contains ?credentials=<filename> parameter, the credentials will
## be stored and looked for in a file named <filename> in this directory.
##
## Valid values: absolute path to a directory
## Default value: /etc/zypp/credentials.d
##
# credentials.global.dir = /etc/zypp/credentials.d

##
## Global credentials catalog file path.
##
## This file contains a catalog of all known user credentials which were
## not stored via the ?credentials=<filename> URL parameter, i.e. passed
## in URL as username:password component, or entered by user in
## an authentication dialog.
##
## Valid values: absolute path to a file
## Default value: /etc/zypp/credentials.cat
##
# credentials.global.file = /etc/zypp/credentials.cat
```