

---

# **dnf-plugins-core Documentation**

*Release 4.4.2-1*

**Tim Lauridsen**

**Sep 21, 2023**



---

## Contents

---

<b>1</b>	<b>Core DNF Plugins Release Notes</b>	<b>3</b>
<b>2</b>	<b>DNF builddep Plugin</b>	<b>23</b>
<b>3</b>	<b>DNF changelog Plugin</b>	<b>25</b>
<b>4</b>	<b>DNF config-manager Plugin</b>	<b>27</b>
<b>5</b>	<b>DNF copr Plugin</b>	<b>29</b>
<b>6</b>	<b>DNF debug Plugin</b>	<b>31</b>
<b>7</b>	<b>DNF debuginfo-install Plugin</b>	<b>33</b>
<b>8</b>	<b>DNF download Plugin</b>	<b>35</b>
<b>9</b>	<b>DNF generate_completion_cache Plugin</b>	<b>37</b>
<b>10</b>	<b>DNF groups-manager Plugin</b>	<b>39</b>
<b>11</b>	<b>DNF leaves Plugin</b>	<b>41</b>
<b>12</b>	<b>DNF local Plugin</b>	<b>43</b>
<b>13</b>	<b>DNF migrate Plugin</b>	<b>45</b>
<b>14</b>	<b>DNF modulesync Plugin</b>	<b>47</b>
<b>15</b>	<b>DNF needs-restarting Plugin</b>	<b>49</b>
<b>16</b>	<b>DNF post-transaction-actions Plugin</b>	<b>51</b>
<b>17</b>	<b>DNF repoclosure Plugin</b>	<b>53</b>
<b>18</b>	<b>DNF repodiff Plugin</b>	<b>55</b>
<b>19</b>	<b>DNF repograph Plugin</b>	<b>57</b>
<b>20</b>	<b>DNF repomanage Plugin</b>	<b>59</b>

<b>21 DNF reposync Plugin</b>	<b>61</b>
<b>22 DNF show-leaves Plugin</b>	<b>63</b>
<b>23 DNF system-upgrade Plugin</b>	<b>65</b>
<b>24 DNF versionlock Plugin</b>	<b>69</b>
<b>25 See Also</b>	<b>73</b>
<b>26 Indices and tables</b>	<b>75</b>

This documents core plugins of DNF:



---

## Core DNF Plugins Release Notes

---

### Contents

- *Core DNF Plugins Release Notes*
  - *4.4.2 Release Notes*
  - *4.4.1 Release Notes*
  - *4.4.0 Release Notes*
  - *4.3.1 Release Notes*
  - *4.3.0 Release Notes*
  - *4.2.1 Release Notes*
  - *4.2.0 Release Notes*
  - *4.1.0 Release Notes*
  - *4.0.24 Release Notes*
  - *4.0.23 Release Notes*
  - *4.0.22 Release Notes*
  - *4.0.21 Release Notes*
  - *4.0.20 Release Notes*
  - *4.0.19 Release Notes*
  - *4.0.18 Release Notes*
  - *4.0.17 Release Notes*
  - *4.0.16 Release Notes*
  - *4.0.15 Release Notes*

- [4.0.14 Release Notes](#)
- [4.0.13 Release Notes](#)
- [4.0.12 Release Notes](#)
- [4.0.11 Release Notes](#)
- [4.0.10 Release Notes](#)
- [4.0.9 Release Notes](#)
- [4.0.8 Release Notes](#)
- [4.0.7 Release Notes](#)
- [4.0.6 Release Notes](#)
- [4.0.4 Release Notes](#)
- [4.0.3 Release Notes](#)
- [4.0.2 Release Notes](#)
- [4.0.0 Release Notes](#)
- [3.0.4 Release Notes](#)
- [3.0.3 Release Notes](#)
- [3.0.2 Release Notes](#)
- [3.0.1 Release Notes](#)
- [2.1.5 Release Notes](#)
- [2.1.4 Release Notes](#)
- [2.1.3 Release Notes](#)
- [2.1.2 Release Notes](#)
- [2.1.1 Release Notes](#)
- [2.1.0 Release Notes](#)
- [2.0.0 Release Notes](#)
- [1.1.0 Release Notes](#)
- [1.0.2 Release Notes](#)
- [1.0.1 Release Notes](#)
- [1.0.0 Release Notes](#)
- [0.1.21 Release Notes](#)
- [0.1.20 Release Notes](#)
- [0.1.18 Release Notes](#)
- [0.1.17 Release Notes](#)
- [0.1.16 Release Notes](#)
- [0.1.15 Release Notes](#)
- [0.1.14 Release Notes](#)



- [0.1.13 Release Notes](#)
- [0.1.12 Release Notes](#)
- [0.1.11 Release Notes](#)
- [0.1.10 Release Notes](#)
- [0.1.9 Release Notes](#)
- [0.1.8 Release Notes](#)
- [0.1.7 Release Notes](#)
- [0.1.6 Release Notes](#)
- [0.1.5 Release Notes](#)
- [0.1.4 Release Notes](#)
- [0.1.3 Release Notes](#)
- [0.1.2 Release Notes](#)
- [0.1.1 Release Notes](#)
- [0.1.0 Release Notes](#)

## 1.1 4.4.2 Release Notes

- Fixed copr.vendor.conf not loading
- “dnf copr enable” on “Asahi Fedora Linux Remix” guesses epel.x86\_64
- system-upgrade: change http to https in unit file
- Bug fixes: - Fix systemd dependencies when using `–poweroff` option in system-upgrade plugin (RhBug:2211844)

Bugs fixed in 4.4.2:

- Bug 2211844 - ‘dnf offline-upgrade reboot `–poweroff`’ doesn’t poweroff after upgrade

## 1.2 4.4.1 Release Notes

- New features: - reposync: Implement `–safe-write-path` option (RhBug:1898089)
- Bug fixes: - needs-restarting: Catch exception when no systemd unit exists for pid (RhBug:2122587) - post-transaction-actions: Fix ConfigParser.substitute call
- Others: - builddep: Avoid using obsolete RPM API - yum-utils: Only depend on python3-dnf, not dnf

Bugs fixed in 4.4.1:

- Bug 1898089 - dnf reposync doesn’t handle well relative paths
- Bug 2122587 - needs-restarting -s fail with mysql container

## 1.3 4.4.0 Release Notes

- New features: - system-upgrade: Move from extras to core (RhBug:2054235) - system-upgrade: Add `--poweroff` option to `reboot` - copr: Switch to reading a `copr.vendor.conf` file to determine a vendor ID - config-manager: Allow to specify the “main” section
- Bug fixes: - system-upgrade: Add support for security filters in `offline-upgrade` (RhBug:1939975) - needs-restarting: Fix boot time derivation for systems with no `rtc` (RhBug:2137935) - download: Skip downloading weak deps when `install_weak_deps=False` - reposync: Documentation update (RhBug:2132383, 2182004)

Bugs fixed in 4.4.0:

- Bug 2054235 - Offline updates
- Bug 1939975 - ‘dnf offline-upgrade’ does not support `--advisory` properly
- Bug 2137935 - dnf needs-restarting always true
- Bug 2132383 - reposync produces wrong metadata when using both `-n` and `--download-metadata` options
- Bug 2182004 - reposync produces wrong metadata when using both `-n` and `--download-metadata` options

## 1.4 4.3.1 Release Notes

- Update translations (fix RhBug:2127011)

Bugs fixed in 4.3.1:

- Bug 2127011 - dnf-plugins-core show errors when use `LANG=es_ES.UTF-8`

## 1.5 4.3.0 Release Notes

- copr: Guess EPEL chroots for CentOS Stream (RhBug:2058471)
- Update translations
- Bug fixes: - builddep: Warning when using macros with source rpms (RhBug:2077820) - [repomanage] Modules are used only when they belong to target repo (RhBug:2072441) - Update documentation for config-manager used with subscription-manager (RhBug:2075366)

Bugs fixed in 4.3.0:

- Bug 2072441 - repomanage `--new` seems to exclude currently enabled modules
- Bug 2077820 - dnf builddep option `--define rpm` macro does not work with `src.rpm`
- Bug 2075366 - unable to enable or disable repositories using yum-config-manager
- Bug 2058471 - dnf copr enable fails to pick epel-9 chroot

## 1.6 4.2.1 Release Notes

- Skip all non rpm tsi for `transaction_action` plugins (rhbug:2023652)

Bugs fixed in 4.2.1:

- Bug 2023652 - `KeyError` after upgrading packages (post-transaction actions)

## 1.7 4.2.0 Release Notes

- New features: - repomanage: Add new option `--oldonly` (RhBug:2034736,2058676)

Bugs fixed in 4.2.0:

- Bug 2034736 - reposync -n downloads more than it should
- Bug 2058676 - repomanage list the same package in both `--old` and `--new` results

## 1.8 4.1.0 Release Notes

- New features: - Add a new subpackage with `modulesync` command. The command downloads packages from modules and/or creates a repository with modular data. (RhBug:1868047) - [groups-manager] Use full NEVRA (not only name) for matching packages (RhBug:2013633) - [repoclosure] Print counts of missing dependencies
- Bug fixes: - [reposync] Do not stop downloading packages on the first error (RhBug:2009894) - [versionlock] Fix: Multiple package-name-spec arguments don't lock correctly (RhBug:2001039) (RhBug:2013324) - [versionlock] Update documentation for adding specific version (RhBug:2013332)

Bugs fixed in 4.1.0:

- Bug 2013633 - [RFE] Add ability to yum-groups-manager to identify packages with version and/or architecture specified in package name
- Bug 2009894 - dnf reposync does not continue on 401 errors
- Bug 2013324 - Multiple package-name-spec arguments don't lock correctly
- Bug 1868047 - [RFE] A module-granularity synchronization tool (similar to reposync)
- Bug 2013332 - Locking a version of a not installed package disallows locking other versions later

## 1.9 4.0.24 Release Notes

- Bug fixes: - [copr] on CentOS Stream, enable centos stream chroot instead of not epel 8 (RhBug:1994154) - [copr] Avoid using deprecated function `distro.linux_distribution()` (RhBug:2011550) - [copr] don't traceback on empty lines in `/etc/os-release`

Bugs fixed in 4.0.24:

- Bug 1994154 - wrong baseurl / dnf copr enable / centos stream
- Bug 2011550 - dnf-plugins-core uses `distro.linux_distribution()`, deprecated in distro 1.6.0

## 1.10 4.0.23 Release Notes

- New features: - [leaves] Show strongly connected components
- Bug fixes: - [needs-restarting] Fix wrong boot time (RhBug:1960437) - [playground] Disable playground command, since it doesn't work

Bugs fixed in 4.0.23:

- Bug 1960437 - needs-restarting calculating wrong boot time

## 1.11 4.0.22 Release Notes

- New features: - [repomanage] Allow running only with metadata - [repomanage] Enhance documentation (RhBug:1898293)
- Bug fixes: - [versionlock] Locking obsoleted package does not make the obsoleter unavailable (RhBug:1957280) - [versionlock] Work correctly with packages with minorbump part of release (RhBug:1961217)

Bugs fixed in 4.0.22:

- Bug 1957280 - DNF with versionlock silences a conflict due to a provide
- Bug 1961217 - procs-ng gets updated regardless of versionlock state
- Bug 1898293 - repomanage `--old` does not list the oldest package per module

## 1.12 4.0.21 Release Notes

- Add missing command line option to documentation
- doc: add packages to needs-restarting conf
- Set blacklist subcommand as deprecated
- Bug fixes: - Bugs fixed (RhBug:1914827,1916782)

Bugs fixed in 4.0.21:

- Bug 1914827 - [RHEL8] dnf reposync implicitly downloads source rpms in spite of no `--source` option

## 1.13 4.0.20 Release Notes

- Bug fixes: - Removed dependency on `dnf.yum.misc.Checksum` class (RhBug:1935465)

Bugs fixed in 4.0.20:

- Bug 1935465 - Cached rpms in dnf uses historic yum checksum code instead of librepo's checksum

## 1.14 4.0.19 Release Notes

- copr: allow only 2 arguments with copr enable command
- [needs-restarting] fix `-r` in nspawn containers (RhBug:1913962,1914251)
- Add `--gpgcheck` option to reposync (RhBug:1856818) (RhBug:1856818)
- Re-introduce yum-groups-manager functionality (RhBug:1826016)
- [repomanage] Don't use cached metadata (RhBug:1899852)
- [needs-restarting] add `-s` to list services (RhBug:1772939) (RhBug:1772939)
- New features: - Add `--gpgcheck` option to reposync (RhBug:1856818) (RhBug:1856818) - Re-introduce yum-groups-manager functionality (RhBug:1826016)
- Bug fixes: - [repomanage] Don't use cached metadata (RhBug:1899852) - [repomanage] Fix ordering of modular stream versions - [needs-restarting] add `-s` to list services (RhBug:1772939) (RhBug:1772939)

Bugs fixed in 4.0.19:

- Bug 1913962 - “dnf needs-restarting -r” work incorrectly inside systemd-nspawn containers
- Bug 1772939 - For “needs-restarting” command. re-Include option (-s --services – “List the affected systemd services only.”)
- Bug 1914251 - “dnf needs-restarting -r” work incorrectly inside systemd-nspawn containers
- Bug 1899852 - repomanage is broken, reports not existing packages
- Bug 1856818 - [RFE] add gpgcheck back to reposync

## 1.15 4.0.18 Release Notes

- [needs-restarting] Fix plugin fail if needs-restarting.d does not exist
- [needs-restarting] add kernel-rt to reboot list
- Fix debug-restore command
- [config-manager] enable/disable comma separated pkgs (RhBug:1830530)
- [debug] Use standard demands.resolving for transaction handling
- [debug] Do not remove install-only packages (RhBug:1844533)
- return error when dnf download failed
- README: Reference Fedora Weblate instead of Zanata
- [reposync] Add latest NEVRAs per stream to download (RhBug: 1833074)
- copr: don't try to list runtime dependencies
- Bug fixes: - Bugs fixed (RhBug:1863433)

Bugs fixed in 4.0.18:

- Bug 1830530 - request to re-introduce functionality - dnf [config-manager] --enable/disablerepo a-repo,b-repo,some\*
- Bug 1863433 - dnf-plugins-core: FTBFS in Fedora rawhide/f33
- Bug 1833074 - reposync --newest-only does not download the latest package
- Bug 1844533 - yum debug-restore removes all but one kernel even though the dump has multiple kernels.
- Bug 1844925 - when i execute “dnf download” failed i want to get return value 1 not 0 in shell.

## 1.16 4.0.17 Release Notes

- [repomanage] Add modular support (RhBug:1804720)
- [needs-restarting] add options using .conf file (RhBug:1810123)

Bugs fixed in 4.0.17:

- Bug 1804720 - repomanage tool is not working correctly with modules in AppStream
- Bug 1810123 - [RFE] Provide ability to add packages to the needs-restarting plugin configuration.

## 1.17 4.0.16 Release Notes

- [versionlock] Take obsoletes into account (RhBug:1627124)
- Move args “--set-enabled”, “--set-disabled” from DNF (RhBug:1727882)
- Add missing arguments --set-enabled/--set-disabled into error message
- Warn when --enablerepo/--disablerepo args were passed (RhBug:1727882)
- [copr] add support for enabling/disabling runtime dependencies
- [copr] no-liability text to be always printed

Bugs fixed in 4.0.16:

- Bug 1727882 - yum-config-manager cannot update repos on RHEL8

## 1.18 4.0.15 Release Notes

- Support remote files in dnf builddep
- [download] Respect repo priority (RhBug:1800342)

Bugs fixed in 4.0.15:

- Bug 1800342 - dnf download and list do not respect repo priority

## 1.19 4.0.14 Release Notes

- Fix conflict for dnf download --resolve (RhBug:1787908)
- config-manager calls parser error when without options (RhBug:1782822)
- Update reposync.py with --norepopath option
- Fix: don't open stdin if versionlock is missing (RhBug:1785563)

Bugs fixed in 4.0.14:

- Bug 1787908 - dnf treats identical package in two repositories (testing/stable) as conflicting request
- Bug 1782822 - dnf config-manager prints nothing and exits with 0
- Bug 1785563 - When using versionlock with -q option ALL outputs are dismissed

## 1.20 4.0.13 Release Notes

- Fix: config\_manager respect config file location during save
- [reposync] Fix --delete with multiple repos (RhBug:1774103)
- Redesign reposync --latest for modular system (RhBug:1775434)
- [doc] Skip creating and installing migrate documentation for Python 3+
- [config-manager] Allow use of --set-enabled without arguments (RhBug:1679213)
- [versionlock] Prevent conflicting/duplicate entries (RhBug:1782052)

Bugs fixed in 4.0.13:

- Bug 1679213 - yum-config-manager `--enable/--disable` without arguments is ambiguous
- Bug 1782052 - dnf versionlock will accept equivalent package names to add or exclude multiple times
- Bug 1775434 - reposync does not properly sync modules with `--newest` option
- Bug 1774103 - The `--delete` option when used with reposync command deletes the packages immediately after download.

## 1.21 4.0.12 Release Notes

- [reposync] Add `--urls` option (RhBug:1686602)
- [versionlock] Add `--raw` option (RhBug:1645564)
- [doc] move manpages for plugins to “dnf-PLUGIN” (RhBug:1706386)
- Add new plugin post-transaction-actions (RhBug:967264)
- [builddep] Add `--skip-unavailable` switch (RhBug:1628634)
- [versionlock] Don't apply excludes on `@System` (RhBug:1726712)
- [reposync] Ignore only modular excludes (RhBug:1750273)

Bugs fixed in 4.0.12:

- Bug 967264 - [plugins] Support post-transaction actions
- Bug 1726712 - Versionlock plugin not allowing upgrades from previous
- Bug 1645564 - [rfe] Allow versionlock to lock only parts of package version
- Bug 1750273 - dnf reposync ignoring `includepkgs` directive (regression)
- Bug 1686602 - [RFE] Add the `--urls` option for RHEL 8 reposync.
- Bug 1706386 - [RFE] Discoverable man page names for DNF plugins

## 1.22 4.0.11 Release Notes

- [spec] Specify attributes for ghost file (RhBug: 1754463)
- download: add the `--debugsource` option (RhBug:1637008)
- Fix incorrect handling richdeps in builddep (RhBug:1756902)

Bugs fixed in 4.0.11:

- Bug 1754463 - dnf-plugins-core package does not verify
- Bug 1637008 - [download] cannot download debugsource (sub)package
- Bug 1756902 - Incorrectly evaluated rich deps

## 1.23 4.0.10 Release Notes

- debuginfo-install: Update both debuginfo and debugsource for updated package (RhBug:1586084)
- copr: Support multilib repofiles (RhBug:1393664)
- copr: Fix disable if copr instance has non-default port
- copr: Fix repoid when using subdirectories in copr project

Bugs fixed in 4.0.10:

- Bug 1689645 - Out of date manual for dnf download
- Bug 1676418 - yum-utils manpage inconsistent with other yum compat manpages
- Bug 1532378 - dnf refuses to install debugsource package in presence of debuginfo package
- Bug 1629412 - debuginfo-install downloads wrong debuginfo

## 1.24 4.0.9 Release Notes

- [spec] Rename dnf-utils to yum-utils
- [builddep] Report all rpm errors (RhBug:1663619,1658292,1724668)
- [config-manager] `--setopt`: Fix crash with “`--save --dump`”
- [config-manager] `--setopt`: Add globs support to repoid
- [config-manager] `--setopt=key=value` is applied only to the main config
- [config-manager] `--setopt` and empty list of repositories (RhBug:1702678)
- [config-manager] `--setopt`: Add check for existence of input repositories

Bugs fixed in 4.0.9:

- Bug 1663619 - dnf builddep fails on glibc.spec
- Bug 1702678 - Settings are not saved with “`yum config-manager --save --setopt=<repoid>.<option>=<value>`”
- Bug 1724668 - dnf builddep fails trying to parse specfile

## 1.25 4.0.8 Release Notes

- [reposync] Enable timestamp preserving for downloaded data (RhBug:1688537)
- [reposync] Download packages from all streams (RhBug:1714788)
- Make yum-copr manpage available (RhBug:1673902)
- [needs-restarting] Add `--reboothint` option (RhBug:1192946) (RhBug:1639468)
- Set the cost of `_dnf_local` repo to 500, to make it preferred to normal repos

Bugs fixed in 4.0.8:

- Bug 1192946 - [RFE] needs-restarting should give a hint if reboot is required
- Bug 1673920 - confusing yum-plugin-changelog documentation
- Bug 1673902 - missing yum-copr man page



- Bug 1707552 - conflicts with yum-plugin-changelog
- Bug 1714788 - Reposync should sync the entire repository to include module information. reposync should download the packages regardless of whether a module is enabled or disabled
- Bug 1688537 - reposync doesn't preserve timestamp from repo being synced

## 1.26 4.0.7 Release Notes

- Fix: copr disable command traceback (RhBug:1693551)
- [doc] state repoid as repo identifier of config-manager (RhBug:1686779)
- Fix download of src when not the latest requested (RhBug:1649627)

Bugs fixed in 4.0.7:

- Bug 1693551 - Problem with dnf copr disable command

## 1.27 4.0.6 Release Notes

- Use improved config parser that preserves order of data
- [leaves] Show multiply satisfied dependencies as leaves
- [download] Fix downloading an rpm from a URL (RhBug:1678582)
- [download] Fix problem with downloading src pkgs (RhBug:1649627)

## 1.28 4.0.4 Release Notes

- [download] Do not download src without `--source` (RhBug:1666648)

Bugs fixed in 4.0.4:

- Bug 1666648 - dnf download command downloads also a srpm

## 1.29 4.0.3 Release Notes

- Add `changelog` plugin that is used for viewing package changelogs
- New option `--metadata-path` option for reposync plugin

Bugs fixed in 4.0.3:

- Bug 1654529 - dnf versionlock will accept NEVRA forms for additions which it then cannot match when deleting

## 1.30 4.0.2 Release Notes

- Added `repodif` command
- copr: fix enabling Rawhide repository

- Add needs-restarting CLI shim
- [reposync] Fix traceback with `--quiet` option
- [versionlock] Accept more pkgspec forms

Bugs fixed in 4.0.2:

- Bug 1643676 - dnf-plugin-versionlock not accept package-spec string from versionlock.list
- Bug 1648649 - dnf reposync with `--quiet` crashes
- Bug 1644563 - backtrace when trying to disable old style Copr repo filename
- Bug 1557340 - config-manager `--add-repo` created a non-valid repo id
- Bug 1628888 - can't enable copr repository using Rawhide

## 1.31 4.0.0 Release Notes

- Enhance documentation
- [repclosure] check every `--pkg` attribute separately
- [repclosure] Now accepts nevra as a argument of `--pkg` option
- [reposync] enhancements (RhBug:1550063,1582152,1550064,1405789,1598068)
- package-cleanup: remove `--oldkernels`
- Download only packages with unique NEVRAs (RhBug:1612874)

Bugs fixed in 4.0.0:

- Bug 1550063 - dnf reposync doesn't download repodata
- Bug 1505035 - dnf autocompletion wrong (seems to lag behind by one dnf transaction)
- Bug 1571622 - dnf repoclosure with `--pkg` option set to non-existing package doesn't show any warning

## 1.32 3.0.4 Release Notes

- [copr] Huge upgrade of copr plugin
- [spec] Disable building python2 modules on Fedora 30+

Bugs fixed in 3.0.4:

- Bug 1626011 - Don't print progress bar to stdout, and when there's no tty (dnf repoquery)
- Bug 1478208 - [rfe] Support for multiple copr instances

## 1.33 3.0.3 Release Notes

- [download] new option `--alldeps` to download all requirements

## 1.34 3.0.2 Release Notes

Bugs fixed in 3.0.2:

- Bug 1603805 - dnf-plugins-core: FTBFS in Fedora rawhide
- Bug 1571251 - dnf download `--source --resolve` does not work

## 1.35 3.0.1 Release Notes

- Enhanced documentation

Bugs fixed in 3.0.1:

- Bug 1576594 - dnf debuginfo-install ignores out of date debuginfo packages
- Bug 1530081 - repoclosure `--best` broken on multilib arches
- Bug 1547897 - Useless error message: not a valid spec file: can't parse specfile
- Bug 1550006 - dnf download `--resolve` fails for java-1.8.0-openjdk
- Bug 1431491 - versionlock: whitelisting packages doesn't work, if they aren't in the repos.
- Bug 1516857 - [debuginfo-install]: No message when a package does not exist
- Bug 1499623 - Mishandling stderr vs stdout (dnf search, dnf repoquery)
- Bug 1489724 - [abrt] dnf: arch(): config.py:907:arch:TypeError: unhashable type: 'list'

## 1.36 2.1.5 Release Notes

Bugs fixed in 2.1.5:

- Bug 1498426 - dnf download does not download RPMs in fedora rawhide container image

## 1.37 2.1.4 Release Notes

- Added four new options for `list` subcommand of `copr` plugin

Bugs fixed in 2.1.4:

- Bug 1476834 - [abrt] dnf: arch(): config.py:908:arch:TypeError: unhashable type: 'list'

## 1.38 2.1.3 Release Notes

Bugs fixed in 2.1.3:

- Bug 1470843 - upgrade to version 26 fails
- Bug 1279001 - [RFE] Missing `dnf --downloadaddir` option
- Bug 1439514 - dnf config-manager `--add-repo` returns 0 on failure

## 1.39 2.1.2 Release Notes

- *DNF copr Plugin* doesn't crash anymore in some circumstances.
- *DNF debuginfo-install Plugin* doesn't install any additional subpackages anymore, previously it was trying to get all dependent packages recursively and install debuginfo packages for them.

Bugs fixed in 2.1.2:

- Bug 1322599 - debuginfo-install installs needless packages

## 1.40 2.1.1 Release Notes

It introduces new behavior of Versionlock plugin where it doesn't apply any excludes in non-transactional operations like *repoquery*, *list*, *info*, etc.

Bugs fixed in 2.1.1:

- Bug 1458446 - Probably a Type-0 in package-cleanup that is inside dnf-utils.

## 1.41 2.1.0 Release Notes

Additional subpackage in 2.1.0:

- Added new subpackage *dnf-utils* that provides binaries originally provided by *yum-utils*.

Bugs fixed in 2.1.0:

- Bug 1381917 - add aliases for each of yum-utils binaries

## 1.42 2.0.0 Release Notes

- Added `DEBUG` plugin from *dnf-plugins-extras*
- Added `LEAVES` plugin from *dnf-plugins-extras*
- Added `LOCAL` plugin from *dnf-plugins-extras*
- Added `MIGRATE` plugin from *dnf-plugins-extras*
- Added `NEEDS RESTARTING` plugin from *dnf-plugins-extras*
- Added `REPOCLOSURE` plugin from *dnf-plugins-extras*
- Added `REPOGRAPH` plugin from *dnf-plugins-extras*
- Added `REPOMANAGE` plugin from *dnf-plugins-extras*
- Added `SHOW LEAVES` plugin from *dnf-plugins-extras*
- Added `VERSIONLOCK` plugin from *dnf-plugins-extras*

## 1.43 1.1.0 Release Notes

- Updated translations
- *DNF builddep Plugin* doesn't check GPG key of src.rpm anymore
- *DNF builddep Plugin* installs dependencies by provides
- *DNF download Plugin* with `--resolve` now downloads all needed packages for transaction

Bugs fixed in 1.1.0:

- Bug 1429087 - Outdated translations
- Bug 1431486 - dnf wants to check gpg key on builddep from src.rpm
- Bug 1332830 - DNF builddep is not consistently installing packages providing capabilities properly
- Bug 1276611 - dnf download `--resolve` should download everytime requested packages

## 1.44 1.0.2 Release Notes

Newly implemented *DNF download Plugin* options `--url` and `--urlprotocol`.

Bugs fixed in 1.0.2:

- Bug 1250115 - RFE: dnf download `--url`

## 1.45 1.0.1 Release Notes

Minor changes in builddep: print errors from RPM SPEC parser

## 1.46 1.0.0 Release Notes

Repoquery and protected\_packages plugins were integrated into DNF.

Bugs fixed in 1.0.0:

- Bug 1361003 - `dnf.plugin.copr (copr.py) cls.config_chroot` returns `None` (affects setting `dist` in `/etc/dnf/plugins/copr.conf`)
- Bug 1360752 - [rfe] provide a way to read out `$basearch`
- Bug 1350604 - RFE: Add 'build-dep' and 'distrosync' aliases to 'builddep' and 'distro-sync'
- Bug 1325350 - DNF Reposync does complete sync/download but doesn't save any packages
- Bug 1303117 - `hy_package_get_requires` should return also "Requires(pre)"
- Bug 1193823 - [RFE] Plugins can't be disabled in config files
- Bug 1260986 - dnf reposync lists repo as wrong argument

## 1.47 0.1.21 Release Notes

Bugfixes, internal improvements.

Bugs fixed in 0.1.21:

- Bug 1335959 - dnf repoquery `--srpm` does not work
- Bug 1279538 - [perf] dnf repoquery `--resolve` is extremely slow
- Bug 1303311 - [weakdeps] `--whatsupplements` not working

## 1.48 0.1.20 Release Notes

Small fixes in COPR plugin, added `get_reposdir` function to `dnfpluginscore` lib.

## 1.49 0.1.18 Release Notes

Changed COPR server adress to the new one.

## 1.50 0.1.17 Release Notes

Added configuration file for *DNF debuginfo-install Plugin*.

Bugs fixed in 0.1.17:

- Bug 1024701 - [rfe] debuginfo-install plugin: add passive feature to keep debuginfo packages in sync
- Bug 1302214 - Translation incomplete of the download plugin

## 1.51 0.1.16 Release Notes

Documented DNF repoquery options `--unneeded` and `--recent`.

Bugs fixed in 0.1.16:

- Bug 1297511 - `noroot` configure causes a traceback when plugins are loaded without a cli
- Bug 1292475 - RFE: dnf repoquery options `--queryformat` and `--latest-limit` should be usable together

## 1.52 0.1.15 Release Notes

Newly implemented *DNF download Plugin* options `--resolve` and `--debuginfo`.

Improved the start-up time of bash completion.

Reviewed documentation.

Bugs fixed in 0.1.15:

- Bug 1283448 - `python3-dnf-plugins-core` update fails with file conflicts
- Bug 1253237 - dnf config-manager `--setopt` options are not saved to repo (`--save` does not work)

## 1.53 0.1.14 Release Notes

Bugs fixed in 0.1.14:

- Bug 1231572 - dnf repoquery `--duplicated` report is different than `package-cleanup --dupes`
- Bug 1265210 - [abrt] dnf: `needs_restarting.py:101:print_cmd:UnicodeDecodeError: 'ascii' codec can't decode byte 0xc3 in position 108: ordinal not in range(128)`
- Bug 1280416 - Copr inserts `@` in the id of the repo, causing it not to work
- Bug 1270091 - `man dnf.plugin.config-manager` does not mention `--dump` in examples
- Bug 1272936 - [abrt] dnf: `lib.py:170:package_source_name:AttributeError: 'NoneType' object has no attribute 'rstrip'`

## 1.54 0.1.13 Release Notes

Kickstart plugin has been moved to `dnf plugins extras` as a separate `python-dnf-plugins-extras-kickstart` package.

Bugs fixed in 0.1.13:

- Bug 1267808 - dnf playground enable doesn't work
- Bug 1264125 - `UnicodeEncodeError` running dnf repoquery
- Bug 1265622 - 'dnf builddep' is not always as 'clever' as `yum-builddep` or `dnf download --source`
- Bug 1159614 - dnf `debuginfo-install kernel-PAE` picks wrong debuginfo package

## 1.55 0.1.12 Release Notes

Added support of globs to `--whatrequires` and `--whatprovides` options.

Bugs fixed in 0.1.12:

- Bug 1249073 - dnf repoquery does not support wildcards in `--whatrequires`

## 1.56 0.1.11 Release Notes

Option `--arch` now accepts more than one architecture.

Introduced select options `--available`, `--extras`, `--installed`, `--upgrades`.

Added ability to use weak dependencies query options in combination with `--tree` switch.

Bugs fixed in 0.1.11:

- Bug 1250114 - dnf download appears to write the destination twice
- Bug 1186381 - [rfe] Please provide an equivalent of `repoquery --archlist`
- Bug 1225784 - dnf download should print message when it does not find requested packages
- Bug 1233728 - dnf repoquery doesn't support multi args
- Bug 1199601 - RFE: repoquery support for `--pkgnarrow`

- Bug 1156778 - dnf repoquery: sort the output alphabetically

## 1.57 0.1.10 Release Notes

*DNF builddep Plugin* was extended by newly added options `--srpm` and `--spec` for specifying the input file.

Implemented `remove` command in *DNF copr Plugin* plugin.

Bugs fixed in 0.1.10:

- Bug 1226663 - dnf bash completion: show version of packages
- Bug 1184930 - [rfe] repoquery: implement weak deps flag options
- Bug 1234099 - reposync unicode error
- Bug 1241126 - builddep: empty run or run on unknown package name doesn't fail
- Bug 1218299 - TypeError: 'error' object does not support indexing
- Bug 1241135 - allow builddep to process file regardless of filename
- Bug 1244125 - [abrt] dnf: gettext.py:222:ugettext:UnicodeDecodeError: 'ascii' codec can't decode byte 0xe6 in position 0: ordinal not in range(128)

## 1.58 0.1.9 Release Notes

DNF repoquery was extended by newly added select options `--srpm`, `--alldeps` and query option `--tree`.

Bugs fixed in 0.1.9:

- Bug 1128425 - RFE: Missing recursive dependency resolution in repoquery
- Bug 1186382 - [rfe] Please provide an equivalent of repoquery `--srpm`
- Bug 1228693 - dnf config-manager `--add-repo` sets permission to 600
- Bug 1186689 - RFE: make dnf repoquery `--whatrequires` behave like yum based repoquery (add `--alldeps` support)
- Bug 1227190 - error enable copr

## 1.59 0.1.8 Release Notes

This release fixes only packaging issues.

## 1.60 0.1.7 Release Notes

All occurrences of *repoId* option were replaced by *repo* to unified repository specification in plugins.

*DNF builddep Plugin* now accepts packages from repositories as arguments and allows users to define RPM macros used during spec files parsing via `-D` option.

Three new options were added to DNF repoquery: *latest-limit*, *unsatisfied* and *resolve*.

Bugs fixed in 0.1.7:



- Bug 1215154 - [PATCH] dnf debuginfo-install: AttributeError: 'NoneType' object has no attribute 'replace'
- Bug 1074585 - [RFE] 'dnf builddep' should be able to take just the package name as argument rather than SPEC or SRPM
- Bug 1156487 - RFE: Add support for repoquery --resolve
- Bug 1208773 - [RFE] Define virtual provides for DNF commands and suggest installing the appropriate one if a command was not found
- Bug 1186948 - [download] data loss: dnf download should not delete files from local repositories

## 1.61 0.1.6 Release Notes

Newly implemented *DNF config-manager Plugin* plugin.

DNF repoquery now accepts *<pkg-spec>*.

Bugs fixed in 0.1.6:

- Bug 1208399 - dnf-3 copr enable plautrba/selinux: NameError: name 'raw\_input' is not defined
- Bug 1194725 - Every dnf-3 transaction returns "Failed loading plugin: copr"
- Bug 1198442 - dnf-plugins-core package does not work with dnf package in rawhide
- Bug 1193047 - [abrt] python3-dnf: copr.py:146:run:AttributeError: 'module' object has no attribute 'FancyURLopener'
- Bug 1196952 - dnf repoquery does not provide -f like yum-utils
- Bug 1171046 - dnf debuginfo-install can't install packages when given exact NVR
- Bug 1179366 - repoquery --whatprovides does not accept wildchars as \*

## 1.62 0.1.5 Release Notes

*DNF builddep Plugin* accepts also *nosrc.rpm* package.

DNF repoquery adds *--list* switch to show files the package contains.

Bugs fixed in 0.1.5:

- Bug 1187773 - [abrt] dnf: copr.py:230:\_download\_repo:NameError: global name 'e' is not defined
- Bug 1178239 - [abrt] dnf: posixpath.py:80:join:UnicodeDecodeError: 'ascii' codec can't decode byte 0xc5 in position 23: ordinal not in range(128)
- Bug 1166126 - dnf builddep does not recognize nosrc.rpm packages
- Bug 1155211 - dnf builddep: print what's missing when package not found

## 1.63 0.1.4 Release Notes

Provides *DNF needs-restarting Plugin* and *DNF reposync Plugin*.

Bugs fixed in 0.1.4:

- Bug 1139738 - [rfe] reposync

- Bug 1144003 - [abrt] dnf: download.py:181:\_get\_query\_source:TypeError: 'NoneType' object has no attribute '\_\_getitem\_\_'

## 1.64 0.1.3 Release Notes

Added info switch to DNF repoquery

Bugs fixed in 0.1.3:

- Bug 1135984 - rfe: add repoquery -i support
- Bug 1134378 - [abrt] dnf: copr.py:147:run:KeyError: 'repos'
- Bug 1123886 - [abrt] dnf: i18n.py:138:\_exact\_width\_char:TypeError: must be unicode, not str

## 1.65 0.1.2 Release Notes

Bugs fixed in 0.1.2:

- Bug 1108321 - debuginfo-install can't install anything if debuginfo with version as installed not found
- Bug 1116389 - dnf copr enable - 404 for repo without builds
- Bug 1118809 - [abrt] dnf: generate\_completion\_cache.py:55:sack:UnicodeDecodeError: 'ascii' codec can't decode byte 0xc3 in position 22: ordinal not in range(128)

## 1.66 0.1.1 Release Notes

Provides protected\_packages and a bugfix to the Copr plugin.

Bugs fixed in 0.1.1:

- Bug 1049310 - Keep running kernel
- Bug 1104088 - Failed loading plugin: copr / ImportError: No module named requests
- Bug 1111855 - DNF needs to protect the basesystem

## 1.67 0.1.0 Release Notes

This release provides the DNF repoquery and a bugfix for the *DNF builddep Plugin*.

Bugs fixed for 0.1.0:

- Bug 1045078 - [rfe] repoquery
- Bug 1103906 - dnf builddep command not working

Install whatever is needed to build the given `.src.rpm`, `.nosrc.rpm` or `.spec` file.

**Warning:** Build dependencies in a package (i.e. `src.rpm`) might be different than you would expect because they were evaluated according macros set on the package build host.

## 2.1 Synopsis

```
dnf builddep <package>...
```

## 2.2 Arguments

**<package>** Either path to `.src.rpm`, `.nosrc.rpm` or `.spec` file or package available in a repository.

## 2.3 Options

All general DNF options are accepted, see *Options* in `dnf(8)` for details.

**--help-cmd** Show this help.

**-D <macro expr>**, **--define <macro expr>** Define the RPM macro named *macro* to the value *expr* when parsing spec files. Does not apply for source rpm files.

**--spec** Treat arguments as `.spec` files.

**--srpm** Treat arguments as source rpm files.

**--skip-unavailable** Skip build dependencies not available in repositories. All available build dependencies will be installed.

Note that *builddep* command does not honor the *-skip-broken* option, so there is no way to skip uninstalleable packages (e.g. with broken dependencies).

## 2.4 Examples

**dnf builddep foobar.spec** Install the needed build requirements, defined in the foobar.spec file.

**dnf builddep --spec foobar.spec.in** Install the needed build requirements, defined in the spec file when filename ends with something different than `.spec`.

**dnf builddep foobar-1.0-1.src.rpm** Install the needed build requirements, defined in the foobar-1.0-1.src.rpm file.

**dnf builddep foobar-1.0-1** Look up foobar-1.0-1 in enabled repositories and install build requirements for its source rpm.

**dnf builddep -D 'scl python27' python-foobar.spec** Install the needed build requirements for the python27 SCL version of python-foobar.

## 3.1 Description

*changelog* is a plugin for viewing package changelogs.

## 3.2 Synopsis

```
dnf changelog [<options>] <package-spec>...
```

## 3.3 Arguments

**<package-spec>** Package specification for packages to display changelogs.

## 3.4 Options

All general DNF options are accepted, see *Options* in *dnf(8)* for details.

**--since=<date>** Show only changelog entries since <date>. To avoid ambiguity using YYYY-MM-DD date format is recommended.

**--count=<number>** Show maximum of <number> changelog entries per package.

**--upgrades** Show only new changelog entries for packages, that provide an upgrade for some of already installed packages.

## 3.5 Examples

Show changelogs for all packages since November 1, 2018:

```
dnf changelog --since=2018-11-1
```

Show 3 latest changelogs of package dnf:

```
dnf changelog --count=3 dnf
```

Show what is new in upgradable packages:

```
dnf changelog --upgrades
```

---

## DNF config-manager Plugin

---

Manage main and repository DNF configuration options, toggle which repositories are enabled or disabled, and add new repositories.

### 4.1 Synopsis

```
dnf config-manager [options] <section>...
```

### 4.2 Arguments

**<section>** This argument can be used to explicitly select the configuration sections to manage. A section can either be `main` or a `repo`. If not specified, the program will select the `main` section and each `repo` used within any `--setopt` options. A `repo` can be specified using globs.

### 4.3 Options

All general DNF options are accepted, see *Options* in *dnf(8)* for details.

**--help-cmd** Show this help.

**--add-repo=URL** Add (and enable) the repo from the specified file or url. If it has to be added into `installroot`, combine it with `--setopt=reposdir=/<installroot>/etc/yum.repos.d` command-line option.

**--dump** Print dump of current configuration values to stdout.

**--set-disabled, --disable** Disable the specified repos (implies `--save`).

**--set-enabled, --enable** Enable the specified repos (implies `--save`).

**--save** Save the current options (useful with `--setopt`).

**--setopt=<option>=<value>** Set a configuration option. To set configuration options for repositories, use `repoId.option` for the `<option>`. Globs are supported in `repoId`.

**Warning:** DNF `config-manager` can misbehave when enabling/disabling repositories generated by tools like `subscription-manager` on RHEL. In this case you should use `subscription-manager` to perform such actions.

## 4.4 Examples

**dnf config-manager --add-repo http://example.com/some/additional.repo** Download `additional.repo` and store it in `repor`.

**dnf config-manager --add-repo http://example.com/different/repo** Create new repo file with `http://example.com/different/repo` as `baseurl` and enable it.

**dnf config-manager --dump** Display main DNF configuration.

**dnf config-manager --dump <section>** Display configuration of a repository identified by `<section>`.

**dnf config-manager --set-enabled <repoId>** Enable repository identified by `<repoId>` and make the change permanent.

**dnf config-manager --set-disabled <repoId1> <repoId2>** Disable repositories identified by `<repoId1>` and `<repoId2>`

**dnf config-manager --set-disabled <repoId1>,<repoId2>** Disable repositories identified by `<repoId1>` and `<repoId2>`

**dnf config-manager --save --setopt=\*.proxy=http://proxy.example.com:3128/ <repo1> <repo2>** Update proxy setting in repositories with `repoId <repo1>` and `<repo2>` and make the change permanent.

**dnf config-manager --save --setopt=\*-debuginfo.gpgcheck=0** Update `gpgcheck` setting in all repositories whose id ends with `-debuginfo` and make the change permanent.



Work with Copr & Playground repositories on the local system.

- The `copr` command is used to add or remove Copr repositories to the local system
- The `playground` is used to enable or disable the Playground repository

## 5.1 Synopsis

```
dnf copr [enable|disable|remove|list|search] <parameters>
```

```
dnf playground [enable|disable|upgrade]
```

## 5.2 Arguments (copr)

**enable name/project [chroot]** Enable the `name/project` Copr repository with the optional `chroot`.

**disable name/project** Disable the `name/project` Copr repository.

**remove name/project** Remove the `name/project` Copr repository.

**list --installed** List installed Copr repositories (default).

**list --enabled** List enabled Copr repositories.

**list --disabled** List disabled Copr repositories.

**list --available-by-user=name** List available Copr repositories for a given name.

**search project** Search for a given `project`.

**enable hub/name/project** Enable the `name/project` Copr repository from the specified Copr hub, Hub is be specified either by its hostname (eg. `copr.fedorainfracloud.org`) or by an ID that's defined in a configuration file.

## 5.3 Options (copr)

All general DNF options are accepted, see *Options* in *dnf(8)* for details.

**--hub Copr** Specify a Copr hub to use. Default is the Fedora Copr: `https://copr.fedorainfracloud.org`.

## 5.4 Configuration (copr)

`/etc/dnf/plugins/copr.conf` `/etc/dnf/plugins/copr.d/` `/usr/share/dnf/plugins/copr.vendor.conf`

**Configuration file should contain a section for each hub, each section having `hostname` (mandatory), `protocol` (default `https`) and `port` (default `443`) parameters.:**

```
[fedora]
hostname = copr.fedorainfracloud.org
protocol = https
port = 443
```

There is also a vendor configuration that allows a vendor to specify the distro ID that copr should use by default. This is useful for vendors that want to use Copr for their own distro. The vendor configuration is in `/usr/share/dnf/plugins/copr.vendor.conf` (optional) or `/etc/dnf/plugins/copr.conf`:

```
[main]
distribution = fedora
releasever = 37
```

## 5.5 Arguments (playground)

**enable** Enable the Playground repository.

**disable** Disable the Playground repository.

**upgrade** Upgrade the Playground repository settings (same as `disable` and then `enable`).

## 5.6 Options (playground)

All general DNF options are accepted, see *Options* in *dnf(8)* for details.

## 5.7 Examples

**copr enable rhsc1/perl516 epel-6-x86\_64** Enable the `rhsc1/perl516` Copr repository, using the `epel-6-x86_64` chroot.

**copr disable rhsc1/perl516** Disable the `rhsc1/perl516` Copr repository

**copr list --available-by-user=rita** List available Copr projects for user `rita`.

**copr search tests** Search for Copr projects named `tests`.

## 6.1 Description

The plugin provides two dnf commands:

**debug-dump** Writes system RPM configuration to a dump file

**debug-restore** Restore the installed packages to the versions written in the dump file. By default, it does not remove already installed versions of install-only packages and only marks those versions that are mentioned in the dump file for installation. The final decision on which versions to keep on the system is left to dnf and can be fine-tuned using the *installonly\_limit* (see *dnf.conf(5)*) configuration option.

---

**Note:** DNF and Yum debug files are not compatible and thus can't be used by the other program.

---

## 6.2 Synopsis

```
dnf debug-dump [--norepos] [<filename>]
dnf debug-restore [--output] [--install-latest] [--ignore-arch]
[--filter-types = [install,remove,replace]] <filename>
```

## 6.3 Arguments

**<filename>** File to write dump to or read from.

## 6.4 Options

All general DNF options are accepted, see *Options* in *dnf (8)* for details.

dnf debug-dump

**--norepos** Do not dump content of enabled repos.

dnf debug-restore

**--filter-types=[install, remove, replace]** Limit package changes to specified type.

**--ignore-arch** When installing package ignore architecture and install missing packages matching the name, epoch, version and release.

**--install-latest** When installing use the latest package of the same name and architecture.

**--output** Only output list of packages which will be installed or removed. No actual changes are done.

**--remove-installonly** Allow removal of install-only packages. Using this option may result in an attempt to remove the running kernel version (in situations when the currently running kernel version is not part of the dump file).

---

## DNF debuginfo-install Plugin

---

Install the associated debuginfo packages for a given package specification.

### 7.1 Synopsis

```
dnf debuginfo-install <pkg-spec>...
```

### 7.2 Arguments

**<pkg-spec>** The package to install the associated debuginfo package for.

### 7.3 Options

All general DNF options are accepted, see *Options* in *dnf(8)* for details.

### 7.4 Configuration

```
/etc/dnf/plugins/debuginfo-install.conf
```

The minimal content of conf file should contain `main` sections with `enabled` and `autoupdate` parameter.

**autoupdate** A boolean option which controls updates of debuginfo packages. If options is enabled and there are debuginfo packages installed it automatically enables all configured debuginfo repositories. (Disabled by default.)

## 7.5 Examples

**dnf debuginfo-install foobar** Install the debuginfo packages for the foobar package.

**dnf upgrade --enablerepo=\*-debuginfo <package-name>-debuginfo** Upgrade debuginfo package of a <package-name>.

**dnf upgrade --enablerepo=\*-debuginfo "\*-debuginfo"** Upgrade all debuginfo packages.

---

## DNF download Plugin

---

Download binary or source packages.

### 8.1 Synopsis

```
dnf download [options] <pkg-spec>...
```

### 8.2 Arguments

**<pkg-spec>** Package specification for the package to download. Local RPMs can be specified as well. This is useful with the `--source` option or if you want to download the same RPM again.

### 8.3 Options

All general DNF options are accepted, see *Options* in *dnf(8)* for details.

**--help-cmd** Show this help.

**--arch <arch>[, <arch>...]** Limit the query to packages of given architectures (default is all compatible architectures with your system). To download packages with arch incompatible with your system use `--forcearch=<arch>` option to change basearch.

**--source** Download the source rpm. Enables source repositories of all enabled binary repositories.

**--debuginfo** Download the debuginfo rpm. Enables debuginfo repositories of all enabled binary repositories.

**--downloaddir** Download directory, default is the current directory (the directory must exist).

**--url** Instead of downloading, print list of urls where the rpms can be downloaded.

**--urlprotocol** Limit the protocol of the urls output by the `-url` option. Options are http, https, rsync, ftp.

**--resolve** Resolves dependencies of specified packages and downloads missing dependencies in the system.

**--alldeps** When used with **--resolve**, download all dependencies (do not skip already installed ones).

## 8.4 Examples

**dnf download dnf** Download the latest dnf package to the current directory.

**dnf download --url dnf** Just print the remote location url where the dnf rpm can be downloaded from.

**dnf download --url --urlprotocols=https --urlprotocols=rsync dnf** Same as above, but limit urls to https or rsync urls.

**dnf download dnf --destdir /tmp/dnl** Download the latest dnf package to the /tmp/dnl directory (the directory must exist).

**dnf download dnf --source** Download the latest dnf source package to the current directory.

**dnf download rpm --debuginfo** Download the latest rpm-debuginfo package to the current directory.

**dnf download btanks --resolve** Download the latest btanks package and the uninstalled dependencies to the current directory.



---

### DNF generate\_completion\_cache Plugin

---

Generates data to speed up shell completion for DNF. The user is not supposed to interact with the plugin in any way.



---

## DNF groups-manager Plugin

---

Create and edit groups repository metadata files.

### 10.1 Synopsis

```
dnf groups-manager [options] [package-name-spec [package-name-spec ...]]
```

### 10.2 Description

groups-manager plugin is used to create or edit a group metadata file for a repository. This is often much easier than writing/editing the XML by hand. The groups-manager can load an entire file of groups metadata and either create a new group or edit an existing group and then write all of the groups metadata back out.

### 10.3 Arguments

**<package-name-spec>** Package to add to a group or remove from a group.

### 10.4 Options

All general DNF options are accepted, see *Options* in *dnf(8)* for details.

**--load=<path\_to\_comps.xml>** Load the groups metadata information from the specified file before performing any operations. Metadata from all files are merged together if the option is specified multiple times.

**--save=<path\_to\_comps.xml>** Save the result to this file. You can specify the name of a file you are loading from as the data will only be saved when all the operations have been performed. This option can also be specified multiple times.

- merge=<path\_to\_comps.xml>** This is the same as loading and saving a file, however the “merge” file is loaded before any others and saved last.
- print** Also print the result to stdout.
- id=<id>** The id to lookup/use for the group. If you don't specify an <id>, but do specify a name that doesn't refer to an existing group, then an id for the group is generated based on the name.
- n <name>, --name=<name>** The name to lookup/use for the group. If you specify an existing group id, then the group with that id will have it's name changed to this value.
- description=<description>** The description to use for the group.
- display-order=<display\_order>** Change the integer which controls the order groups are presented in, for example in `dnf grouplist`.
- translated-name=<lang:text>** A translation of the group name in the given language. The syntax is `lang:text`. Eg. `en:my-group-name-in-english`
- translated-description=<lang:text>** A translation of the group description in the given language. The syntax is `lang:text`. Eg. `en:my-group-description-in-english`.
- user-visible** Make the group visible in `dnf grouplist` (this is the default).
- not-user-visible** Make the group not visible in `dnf grouplist`.
- mandatory** Store the package names specified within the mandatory section of the specified group, the default is to use the default section.
- optional** Store the package names specified within the optional section of the specified group, the default is to use the default section.
- remove** Instead of adding packages remove them. Note that the packages are removed from all sections (default, mandatory and optional).
- dependencies** Also include the names of the direct dependencies for each package specified.

List installed packages not required by any other installed package.

### 11.1 Synopsis

`dnf leaves`

### 11.2 Description

*leaves* lists all the packages installed on your system which are not required as a dependency of another installed package. However two or more packages might depend on each other in a dependency cycle. Packages in such cycles, which are not required by any other package, are also listed.

### 11.3 Options

All general DNF options are accepted, see *Options* in *dnf(8)* for details.

### 11.4 Why is this useful?

The list gives you a nice overview of what is installed on your system without flooding you with anything required by the packages already shown. The following list of arguments basically says the same thing in different ways:

- All the packages on this list is either needed by you, other users of the system or not needed at all – if it was required by another installed package it would not be on the list.
- If you want to uninstall anything from your system (without breaking dependencies) it must involve at least one package on this list.

- If there is anything installed on the system which is not needed it must be on this list – otherwise it would be required as a dependency by another package.

### 12.1 Description

Automatically copy all downloaded packages to a repository on the local filesystem and generating repo metadata.

---

**Note:** Generating repo metadata will work only if you have installed `createrepo_c` package.

---

### 12.2 Configuration

`/etc/dnf/plugins/local.conf`

The minimal content of conf file should contain `main` and `createrepo` sections with `enabled` parameter, otherwise plugin will not work.:

```
[main]
enabled = true

[createrepo]
enabled = true
```

For `main` section you can specify `reporoot` parameter which sets path to local repository.

Other options and comments you can find in configuration file.





### 13.1 Description

Migrates yum's history, group and yumdb data to dnf.

### 13.2 Synopsis

```
dnf migrate
```

### 13.3 Commands

**dnf migrate** Migrate all the data including history, yumdb and groups (if available).

### 13.4 Options

All general DNF options are accepted, see *Options* in *dnf(8)* for details.



Download packages from modules and/or create a repository with modular data.

## 14.1 Synopsis

```
dnf modulesync [options] [<module-spec>...]
```

## 14.2 Description

*modulesync* downloads packages from modules according to provided arguments and creates a repository with modular data in working directory. In environment with modules it is recommend to use the command for redistribution of packages, because DNF does not allow installation of modular packages without modular metadata on the system (Fail-safe mechanism). The command without an argument creates a repository like *createrepo\_c* but with modular metadata collected from all available repositories.

See examples.

## 14.3 Arguments

**<module-spec>** Module specification for the package to download. The argument is an optional.

## 14.4 Options

All general DNF options are accepted. Namely, the `--destdir` option can be used to specify directory where packages will be downloaded and the new repository created. See *Options* in *dnf(8)* for details.

**-n, --newest-only** Download only packages from the newest modules.

**--enable\_source\_repos** Enable repositories with source packages  
**--enable\_debug\_repos** Enable repositories with debug-info and debug-source packages  
**--resolve** Resolve and download needed dependencies

## 14.5 Examples

**dnf modulesync nodejs** Download packages from *nodejs* module and create a repository with modular metadata in working directory

```
dnf download nodejs
```

**dnf modulesync** The first *download* command downloads *nodejs* package into working directory. In environment with modules *nodejs* package can be a modular package therefore when I create a repository I have to insert also modular metadata from available repositories to ensure 100% functionality. Instead of *createrepo\_c* use *dnf modulesync* to create a repository in working directory with *nodejs* package and modular metadata.

**dnf --destdir=/tmp/my-temp modulesync nodejs:14/minimal --resolve** Download package required for installation of *minimal* profile from module *nodejs* and stream *14* into directory */tmp/my-temp* and all required dependencies. Then it will create a repository in */tmp/my-temp* directory with previously downloaded packages and modular metadata from all available repositories.

```
dnf module install nodejs:14/minimal --downloadonly --destdir=/tmp/my-temp
```

**dnf modulesync --destdir=/tmp/my-temp** The first *dnf module install* command downloads package required for installation of *minimal* profile from module *nodejs* and stream *14* into directory */tmp/my-temp*. The second command *dnf modulesync* will create a repository in */tmp/my-temp* directory with previously downloaded packages and modular metadata from all available repositories. In comparison to *dnf --destdir=/tmp/my-temp modulesync nodejs:14/minimal --resolve* it will only download packages required for installation on current system.

## 14.6 See Also

- *dnf(8)*, DNF Command Reference

---

## DNF needs-restarting Plugin

---

Check for running processes that should be restarted.

### 15.1 Synopsis

```
dnf needs-restarting [-u] [-r] [-s]
```

### 15.2 Description

*needs-restarting* looks through running processes and tries to detect those that use files from packages that have been updated after the given process started. Such processes are reported by this tool.

Note that in most cases a process should survive update of its binary and libraries it is using without requiring to be restarted for proper operation. There are however specific cases when this does not apply. Separately, processes often need to be restarted to reflect security updates.

### 15.3 Options

All general DNF options are accepted, see *Options* in *dnf(8)* for details.

**-u, --useronly** Only consider processes belonging to the running user.

**-r, --reboothint** Only report whether a reboot is required (exit code 1) or not (exit code 0).

**-s, --services** Only list the affected systemd services.

### 15.4 Configuration

```
/etc/dnf/plugins/needs-restarting.d/
```

```
/etc/dnf/plugins/needs-restarting.d/pkgname.conf
```

Packages can be added to `needs-restarting` via `conf` files in `config` directory. `Conf` files need to have `.conf` extension or will be ignored.

More than one package is allowed in each file (one package per line) although it is advised to use one file for each package.

Example:

```
echo "dwm" > /etc/dnf/plugins/needs-restarting.d/dwm.conf
```

---

## DNF post-transaction-actions Plugin

---

### 16.1 Description

The plugin allows to define actions to be executed upon completing an RPM transaction. Each action may define a (glob-like) filtering rule on the package NEVRA or package files, as well as whether the package was installed or removed. Actions are defined in action files.

### 16.2 Configuration

The plugin configuration is in `/etc/dnf/plugins/post-transaction-actions.conf`. All configuration options are in the `[main]` section.

**enabled** Whether the plugin is enabled. Default value is `True`.

**actiondir** Path to the directory with action files. Action files must have the “.action” extension. Default value is `“/etc/dnf/plugins/post-transaction-actions.d/”`.

### 16.3 Action file format

Empty lines and lines that start with a ‘#’ character are ignored. Each non-comment line defines an action and consists of three items separated by colons: `package_filter:transaction_state:command`.

**package\_filter** A (glob-like) filtering rule applied on the package NEVRA (also in the shortened forms) or package files.

**transaction\_state** Filters packages according to their state in the transaction.

- `in` - packages that appeared on the system (downgrade, install, obsolete, reinstall, upgrade)
- `out` - packages that disappeared from the system (downgraded, obsoleted, remove, upgraded)
- `any` - all packages

**command** Any shell command. The following variables in the command will be substituted:

- `${name}`, `$name` - package name
- `${arch}`, `$arch` - package arch
- `${ver}`, `$ver` - package version
- `${rel}`, `$rel` - package release
- `${epoch}`, `$epoch` - package epoch
- `${repoid}`, `$repoid` - package repository id
- **`$(state)`, `$state` - the change of package state in the transaction:** “downgrade”, “downgraded”, “install”, “obsolete”, “obsoleted”, “reinstall”, “reinstalled”, “remove”, “upgrade”, “upgraded”

The shell command will be evaluated for each package that matched the `package_filter` and the `transaction_state`. However, after variable substitution, any duplicate commands will be removed and each command will only be executed once per transaction. The order of execution of the commands follows the order in the action files, but may differ from the order of packages in the transaction. In other words, when you define several action lines for the same `package_filter` these lines will be executed in the order they were defined in the action file when the `package_filter` matches a package during the `transaction_state`. However, the order of when a particular `package_filter` is invoked depends on the position of the corresponding package in the transaction.

### 16.3.1 An example action file:

```
# log all packages (state, nevra, repo) in transaction into a file.
*:any:echo '${state} ${name}-${epoch}:${ver}-${rel}.${arch} repo ${repoid}' >>/tmp/
↳post-trans-actions-trans.log

# The same shell command (after variables substitution) is executed only once per
↳transaction.
*:any:echo '${repoid}' >>/tmp/post-trans-actions-repos
# will write each repo only once to /tmp/post-trans-actions-repos, even if multiple
↳packages from
# the same repo were matched
```



Display a list of unresolved dependencies for repositories.

## 17.1 Synopsis

```
dnf repoclosure [<options>]
```

## 17.2 Description

*repoclosure* is a program that reads package metadata from one or more repositories, checks all dependencies, and displays a list of packages with unresolved dependencies.

### 17.2.1 Options

All general DNF options are accepted, see *Options* in *dnf(8)* for details.

- arch <arch>** Query only packages for specified architecture, can be specified multiple times (default is all compatible architectures with your system). To run repoclosure for arch incompatible with your system use `--forcearch=<arch>` option to change basearch.
- best** Check only the newest packages per arch.
- check <repo id>** Specify repo ids to check, can be specified multiple times (default is all enabled).
- newest** Check only the newest packages in the repos.
- pkg <pkg-spec>** Check closure for this package only.
- repo <repo id>** Specify repo ids to query, can be specified multiple times (default is all enabled).

## 17.3 Examples

Display list of unresolved dependencies for all enabled repositories:

```
dnf repoclosure
```

Display list of unresolved dependencies for rawhide repository and packages with architecture noarch and x86\_64:

```
dnf repoclosure --repo rawhide --arch noarch --arch x86_64
```

Display list of unresolved dependencies for zmap package from rawhide repository:

```
dnf repoclosure --repo rawhide --pkg zmap
```

Display list of unresolved dependencies for myrepo, an add-on for the rawhide repository:

```
dnf repoclosure --repo rawhide --check myrepo
```

Display a list of differences between two or more repositories

## 18.1 Synopsis

```
dnf repodiff [<options>]
```

## 18.2 Description

*repodiff* is a program which will list differences between two sets of repositories. Note that by default only source packages are compared.

### 18.2.1 Options

All general DNF options are accepted, see *Options* in *dnf(8)* for details.

**--repo-old=<repoid>, -o <repoid>** Add a <repoid> as an old repository. It is possible to be used in conjunction with **--repofrompath** option. Can be specified multiple times.

**--repo-new=<repoid>, -n <repoid>** Add a <repoid> as a new repository. Can be specified multiple times.

**--archlist=<arch>, -a <arch>** Add architectures to change the default from just comparing source packages. Note that you can use a wildcard "\*" for all architectures. Can be specified multiple times.

**--size, -s** Output additional data about the size of the changes.

**--compare-arch** Normally packages are just compared based on their name, this flag makes the comparison also use the arch. So `foo.noarch` and `foo.x86_64` are considered to be a different packages.

**--simple** Output a simple one line message for modified packages.

**--downgrade** Split the data for modified packages between upgraded and downgraded packages.

## 18.3 Examples

Compare source pkgs in two local repos:

```
dnf repodiff --repofrompath=o,/tmp/repo-old --repofrompath=n,/tmp/repo-new --repo-  
↳old=o --repo-new=n
```

Compare x86\_64 compat. binary pkgs in two remote repos, and two local one:

```
dnf repodiff --repofrompath=o,http://example.com/repo-old --repofrompath=n,http://  
↳example.com/repo-new --repo-old=o --repo-new=n --archlist=x86_64
```

Compare x86\_64 compat. binary pkgs, but also compare architecture:

```
dnf repodiff --repofrompath=o,http://example.com/repo-old --repofrompath=n,http://  
↳example.com/repo-new --repo-old=o --repo-new=n --archlist=x86_64 --compare-arch
```

Output a full package dependency graph in dot format.

## 19.1 Synopsis

```
dnf repograph [<options>] dnf repo-graph [<options>]
```

## 19.2 Description

*repograph* is a program that generates a full package dependency list from a repository and outputs it in dot format.

### 19.2.1 Options

All general DNF options are accepted, see *Options* in *dnf (8)* for details.

**--repo <repoid>** Specify repo ids to query, can be specified multiple times (default is all enabled).

## 19.3 Examples

Output dependency list from all enabled repositories:

```
dnf repograph
```

Output dependency list from rawhide repository:

```
dnf repograph --repoid rawhide
```

Output dependency list from rawhide and koji repository:

```
dnf repo-graph --repo-id rawhide --repo-id koji
```

Manage a repository or a simple directory of rpm packages.

## 20.1 Synopsis

```
dnf repomanage [<optional-options>] [<options>] <path>
```

## 20.2 Description

*repomanage* prints newest or older packages in a repository specified by <path> for easy piping to xargs or similar programs. In case <path> doesn't contain a valid repodata, it is searched for rpm packages which are then used instead. If the repodata are present, *repomanage* uses them as the source of truth, it doesn't verify that they match the present rpm packages. In fact, *repomanage* can run with just the repodata, no rpm packages are needed.

In order to work correctly with modular packages, <path> has to contain repodata with modular metadata. If modular content is present, *repomanage* prints packages from newest or older stream versions in addition to newest or older non-modular packages.

### 20.2.1 Options

All general DNF options are accepted, see *Options* in *dnf(8)* for details.

The following options set what packages are displayed. These options are mutually exclusive, i.e. only one can be specified. If no option is specified, the newest packages are shown.

**--old** Show older packages (for a package or a stream show all versions except the newest one).

**--oldonly** Show older packages (same as `--old`, but exclude the newest packages even when it's included in the older stream versions).

**--new** Show newest packages.

The following options control how packages are displayed in the output:

**-s, --space** Print resulting set separated by space instead of newline.

**-k <keep-number>, --keep <keep-number>** Limit the resulting set to newest <keep-number> packages.

## 20.3 Examples

Display newest packages in current repository (directory):

```
dnf repomanage --new .
```

Display 2 newest versions of each package in “home” directory:

```
dnf repomanage --new --keep 2 ~/
```

Display oldest packages separated by space in current repository (directory):

```
dnf repomanage --old --space .
```



Synchronize packages of a remote DNF repository to a local directory.

## 21.1 Synopsis

```
dnf reposync [options]
```

## 21.2 Description

*reposync* makes local copies of remote repositories. Packages that are already present in the local directory are not downloaded again.

## 21.3 Options

All general DNF options are accepted. Namely, the `--repo` option can be used to specify the repositories to synchronize. See *Options* in *dnf(8)* for details.

**-a <architecture>**, **--arch=<architecture>** Download only packages of given architectures (default is all architectures). Can be used multiple times.

**--delete** Delete local packages no longer present in repository.

**--download-metadata** Download all repository metadata. Downloaded copy is instantly usable as a repository, no need to run *createrepo\_c* on it. When the option is used with *-newest-only*, only latest packages will be downloaded, but metadata will still contain older packages. It might be useful to update metadata using *createrepo\_c -update* to remove the packages with missing RPM files from metadata. Otherwise, DNF ends with an error due to the missing files whenever it tries to download older packages.

- g, --gpgcheck** Remove packages that fail GPG signature checking after downloading. Exit code is 1 if at least one package was removed. Note that for repositories with `gpgcheck=0` set in their configuration the GPG signature is not checked even with this option used.
- m, --downloadcomps** Also download and uncompress `comps.xml`. Consider using `--download-metadata` option which will download all available repository metadata.
- metadata-path** Root path under which the downloaded metadata are stored. It defaults to `--download-path` value if not given.
- n, --newest-only** Download only newest packages per-repo.
- norepopath** Don't add the reponame to the download path. Can only be used when syncing a single repository (default is to add the reponame).
- p <download-path>, --download-path=<download-path>** Root path under which the downloaded repositories are stored, relative to the current working directory. Defaults to the current working directory. Every downloaded repository has a subdirectory named after its ID under this path.
- safe-write-path** Specify the filesystem path prefix under which the reposync is allowed to write. If not specified it defaults to download path of the repository. Useful for repositories that use relative locations of packages out of repository directory (e.g. `./packages_store/foo.rpm`). Use with care, any file under the `safe-write-path` can be overwritten. Can be only used when syncing a single repository.
- remote-time** Try to set the timestamps of the downloaded files to those on the remote side.
- source** Download only source packages.
- u, --urls** Just print urls of what would be downloaded, don't download.

## 21.4 Examples

**dnf reposync --repoid=the\_repo** Synchronize all packages from the repository with id “the\_repo”. The synchronized copy is saved in “the\_repo” subdirectory of the current working directory.

**dnf reposync -p /my/repos/path --repoid=the\_repo** Synchronize all packages from the repository with id “the\_repo”. In this case files are saved in “/my/repos/path/the\_repo” directory.

**dnf reposync --repoid=the\_repo --download-metadata** Synchronize all packages and metadata from “the\_repo” repository.

Repository synchronized with `--download-metadata` option can be directly used in DNF for example by using `--repofrompath` option:

```
dnf --repofrompath=syncedrepo,the_repo --repoid=syncedrepo list --available
```

## 21.5 See Also

- *dnf(8)*, DNF Command Reference

---

### DNF show-leaves Plugin

---

Shows newly installed leaf packages and packages that became leaves after a transaction.

For more information about leaf packages, see documentation for the `leaves` DNF plugin.



---

## DNF system-upgrade Plugin

---

### 23.1 Description

DNF system-upgrades plugin provides three commands: `system-upgrade`, `offline-upgrade`, and `offline-distro-sync`. Only `system-upgrade` command requires increase of distribution major version (`--releasever`) compared to installed version.

`dnf system-upgrade` is a recommended way to upgrade a system to a new major release. It replaces `fedup` (the old Fedora Upgrade tool). Before you proceed ensure that your system is fully upgraded (`dnf --refresh upgrade`).

The `system-upgrade` command also performs additional actions necessary for the upgrade of the system, for example an upgrade of groups and environments.

### 23.2 Synopsis

```
dnf system-upgrade download --releasever VERSION [OPTIONS]
dnf system-upgrade reboot
dnf system-upgrade reboot --poweroff
dnf system-upgrade clean
dnf system-upgrade log
dnf system-upgrade log --number=<number>
dnf offline-upgrade download [OPTIONS]
dnf offline-upgrade reboot
dnf offline-upgrade reboot --poweroff
dnf offline-upgrade clean
dnf offline-upgrade log
```

```
dnf offline-upgrade log --number=<number>
dnf offline-distro-sync download [OPTIONS]
dnf offline-distro-sync reboot
dnf offline-distro-sync reboot --poweroff
dnf offline-distro-sync clean
dnf offline-distro-sync log
dnf offline-distro-sync log --number=<number>
```

## 23.3 Subcommands

**download** Downloads everything needed to upgrade to a new major release.

**reboot** Prepares the system to perform the upgrade, and reboots to start the upgrade. This can only be used after the `download` command completes successfully.

**clean** Remove previously-downloaded data. This happens automatically at the end of a successful upgrade.

**log** Used to see a list of boots during which an upgrade was attempted, or show the logs from an upgrade attempt. The logs for one of the boots can be shown by specifying one of the numbers in the first column. Negative numbers can be used to number the boots from last to first. For example, `log --number=-1` can be used to see the logs for the last upgrade attempt.

## 23.4 Options

**--releasever=VERSION REQUIRED.** The version to upgrade to. Sets `$releasever` in all enabled repos. Usually a number, or `rawhide`.

**--downloadaddir=<path>** Redirect download of packages to provided `<path>`. By default, packages are downloaded into (per repository created) subdirectories of `/var/lib/dnf/system-upgrade`.

**--distro-sync** Behave like `dnf distro-sync`: always install packages from the new release, even if they are older than the currently-installed version. This is the default behavior.

**--no-downgrade** Behave like `dnf update`: do not install packages from the new release if they are older than what is currently installed. This is the opposite of `--distro-sync`. If both are specified, the last option will be used. The option cannot be used with the `offline-distro-sync` command.

**--poweroff** When applied with the `reboot` subcommand, the system will power off after upgrades are completed, instead of restarting.

**--number** Applied with `log` subcommand will show the log specified by the number.

## 23.5 Notes

`dnf system-upgrade reboot` does not create a “System Upgrade” boot item. The upgrade will start regardless of which boot item is chosen.

The `DNF_SYSTEM_UPGRADE_NO_REBOOT` environment variable can be set to a non-empty value to disable the actual reboot performed by `system-upgrade` (e.g. for testing purposes).

Since this is a DNF plugin, options accepted by `dnf` are also valid here, such as `--allowerasing`. See `dnf(8)` for more information.

The `fedup` command is not provided, not even as an alias for `dnf system-upgrade`.

## 23.6 Bugs

Upgrading from install media (e.g. a DVD or .iso file) currently requires the user to manually set up a DNF repo and `fstab` entry for the media.

## 23.7 Examples

### 23.7.1 Typical upgrade usage

```
dnf --refresh upgrade
dnf system-upgrade download --releasever 26
dnf system-upgrade reboot
```

### 23.7.2 Show logs from last upgrade attempt

```
dnf system-upgrade log --number=-1
```

## 23.8 Reporting Bugs

Bugs should be filed here:

<https://bugzilla.redhat.com/>

For more info on filing bugs, see the Fedora Project wiki:

[https://fedoraproject.org/wiki/How\\_to\\_file\\_a\\_bug\\_report](https://fedoraproject.org/wiki/How_to_file_a_bug_report)

[https://fedoraproject.org/wiki/Bugs\\_and\\_feature\\_requests](https://fedoraproject.org/wiki/Bugs_and_feature_requests)

Please include `/var/log/dnf.log` and the output of `dnf system-upgrade log --number=-1` (if applicable) in your bug reports.

Problems with dependency solving during download are best reported to the maintainers of the package(s) with the dependency problems.

Similarly, problems encountered on your system after the upgrade completes should be reported to the maintainers of the affected components. In other words: if (for example) KDE stops working, it's best if you report that to the KDE maintainers.

## 23.9 See Also

`dnf(8)`, `dnf.conf(5)`, `journalctl(1)`.

### 23.9.1 Project homepage

<https://github.com/rpm-software-management/dnf-plugins-core>

## 23.10 Authors

Will Woods <[wwoods@redhat.com](mailto:wwoods@redhat.com)>

Štěpán Smetana <[ssmetana@redhat.com](mailto:ssmetana@redhat.com)>



## 24.1 Description

*versionlock* is a plugin that takes a set of names and versions for packages and excludes all other versions of those packages. This allows you to protect packages from being updated by newer versions. Alternately, it accepts a specific package version to exclude from updates, e.g. for when it's necessary to skip a specific release of a package that has known issues.

The plugin provides a command *versionlock* which allows you to view and edit the list of locked packages easily.

The plugin will walk each line of the versionlock file, and parse out the name and version of the package. It will then exclude any package by that name that doesn't match one of the versions listed within the file. This is basically the same as using *dnf -exclude* for the package name itself (as you cannot exclude installed packages), but *dnf* will still see the versions you have installed/versionlocked as available so that *dnf reinstall* will still work, etc.

It can also work in the opposite way, like a fast exclude, by prefixing a '!' character to the version recorded in the lock list file. This specifically excludes a package that matches the version exactly.

Note the versionlock plugin does not apply any excludes in non-transactional operations like *repoquery*, *list*, *info*, etc.

## 24.2 Synopsis

```
dnf versionlock [options] [add|exclude|list|delete|clear]
[<package-name-spec>]
```

## 24.3 Arguments

**<package-name-spec>** Package spec to lock or exclude.

## 24.4 Subcommands

**dnf versionlock add <package-name-spec>** Add a versionlock for all available packages matching the spec. It means that only versions of packages represented by <package-name-spec> will be available for transaction operations. Each <package-name-spec> is converted to concrete NEVRAs which are used for locking. The NEVRAs to lock to are first searched among installed packages and then (if none is found) in all currently available packages.

Examples:

Locking a package to the version installed:

```
$ dnf repoquery --installed bash
bash-0:5.0.7-1.fc30.x86_64

$ dnf repoquery bash
bash-0:5.0.2-1.fc30.i686
bash-0:5.0.2-1.fc30.x86_64
bash-0:5.0.7-1.fc30.i686
bash-0:5.0.7-1.fc30.x86_64

$ dnf versionlock add bash
Adding versionlock on: bash-0:5.0.7-1.fc30.*
```

Locking not installed package to any of available versions:

```
$ dnf repoquery --installed mutt

$ dnf repoquery mutt
mutt-5:1.11.4-1.fc30.x86_64
mutt-5:1.12.1-3.fc30.x86_64

$ dnf versionlock add mutt
Adding versionlock on: mutt-5:1.11.4-1.fc30.*
Adding versionlock on: mutt-5:1.12.1-3.fc30.*
```

---

**Note:** Be careful when adding specific versions

If you add a package specifying a version with `dnf versionlock mutt-5:1.11.4-1.fc30.x86_64` then, if you run `dnf versionlock add mutt versionlock` will not add `mutt-5:1.12.1-3.fc30.x86_64`.

---

**dnf versionlock exclude <package-name-spec>** Add an exclude (within versionlock) for the available packages matching the spec. It means that packages represented by <package-name-spec> will be excluded from transaction operations.

**dnf versionlock list** or **dnf versionlock** List the current versionlock entries.

**dnf versionlock delete <package-name-spec>** Remove any matching versionlock entries.

**dnf versionlock clear** Remove all versionlock entries.

## 24.5 Options

All general DNF options are accepted, see *Options* in *dnf(8)* for details.

**--raw** Do not resolve `<package-name-spec>` to NEVRAs to find specific version to lock to. Instead `<package-name-spec>` are used as they are. This enables locking to not yet available versions of the package. For example you may want to keep the `bash` package on major version 5 and consume any future updates as far as they keep the major version:

```
$ dnf versionlock add --raw 'bash-5.*'  
Adding versionlock on: bash-5.*
```

## 24.6 Configuration

`/etc/dnf/plugins/versionlock.conf`

The minimal content of conf file should contain main sections with `enabled` and `locklist` parameters.

**locklist** This option is a string that points to the file which has the versionlock information in it. Note that the file has to exist (or the versionlock plugin will make dnf exit). However, it can be empty.

The file takes entries in the format of `<package-name-spec>` (optionally prefixed with `!` for excludes). See *Specifying packages* in `dnf(8)` for details.

## 24.7 Notes

A specified package does not have to exist within the available cache of repository data to be considered valid for locking or exclusion. This is by design, to accommodate use cases such as a presently disabled repository. However, a package must exist in the repository cache when the `add` or `exclude` subcommands are invoked for it.



## CHAPTER 25

---

### See Also

---

- [DNF project homepage](#)
- [Core DNF Plugins project homepage](#)
- [DNF documentation](#)
- [DNF Plugins Extras documentation](#)
- [Hawkey documentation](#)



## CHAPTER 26

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`