

---

# **dnf-plugins-core Documentation**

*Release 2.1.1-1*

**Tim Lauridsen**

**Jun 21, 2017**



---

## Contents

---

<b>1</b>	<b>Core DNF Plugins Release Notes</b>	<b>3</b>
<b>2</b>	<b>DNF builddep Plugin</b>	<b>11</b>
<b>3</b>	<b>DNF config-manager Plugin</b>	<b>13</b>
<b>4</b>	<b>DNF copr Plugin</b>	<b>15</b>
<b>5</b>	<b>DNF debug Plugin</b>	<b>17</b>
<b>6</b>	<b>DNF debuginfo-install Plugin</b>	<b>19</b>
<b>7</b>	<b>DNF download Plugin</b>	<b>21</b>
<b>8</b>	<b>DNF generate_completion_cache Plugin</b>	<b>23</b>
<b>9</b>	<b>DNF leaves Plugin</b>	<b>25</b>
<b>10</b>	<b>DNF local Plugin</b>	<b>27</b>
<b>11</b>	<b>DNF migrate Plugin</b>	<b>29</b>
<b>12</b>	<b>DNF needs-restarting Plugin</b>	<b>31</b>
<b>13</b>	<b>DNF repoclosure Plugin</b>	<b>33</b>
<b>14</b>	<b>DNF repograph Plugin</b>	<b>35</b>
<b>15</b>	<b>DNF repomanage Plugin</b>	<b>37</b>
<b>16</b>	<b>DNF reposync Plugin</b>	<b>39</b>
<b>17</b>	<b>DNF show-leaves Plugin</b>	<b>41</b>
<b>18</b>	<b>DNF versionlock Plugin</b>	<b>43</b>
<b>19</b>	<b>See Also</b>	<b>45</b>
<b>20</b>	<b>Indices and tables</b>	<b>47</b>



This documents core plugins of DNF:



---

## Core DNF Plugins Release Notes

---

### Contents

- *Core DNF Plugins 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](#)

## 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.

## 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

## 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`

## 1.1.0 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

## 1.0.2 Release Notes

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

Bugs fixed in 1.0.2:

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

## 1.0.1 Release Notes

Minor changes in `builddep`: print errors from RPM SPEC parser

## 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

## 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

## 0.1.20 Release Notes

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

## 0.1.18 Release Notes

Changed COPR server adress to the new one.

## 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

## 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

## 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)

## 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'`

## 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

## 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`

## 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

## 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)

## 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

## 0.1.8 Release Notes

This release fixes only packaging issues.

## 0.1.7 Release Notes

All occurrences of *reposit* 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

## 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 \*

## 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

## 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\_\_'

## 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

## 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)

## 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

## 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

---

## DNF builddep Plugin

---

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.

### Synopsis

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

### Arguments

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

### Options

**--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.

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

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

## 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.



---

## DNF config-manager Plugin

---

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

### Synopsis

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

### Arguments

**<repo>** Display / modify specified repository. If not specified display / modify main DNF configuration. Repositories can be specified using globs.

### Options

**--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 the specified repos (automatically saves).

**--set-enabled** Enable the specified repos (automatically saves).

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

## Examples

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

**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 repo --dump** Display configuration of repo.

**dnf config-manager --set-enabled repo** Enable repo and make the change permanent.

**dnf config-manager --setopt proxy=http://proxy.example.com:3128/ repo1 repo2 --save** Update proxy setting in repo1 and repo2 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

## Synopsis

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

## 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 name** List available Copr repositories for a given name.

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

## Arguments (playground)

**enable** Enable the Playground repository.

**disable** Disable the Playground repository.

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

## 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 rita** List available Copr projects for user rita.

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

## Description

Writes system RPM configuration to a dump file and restore it.

---

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

---

## Synopsis

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

## Arguments

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

## Options

```
dnf debug-dump
```

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

```
dnf debug-restore
```

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

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

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

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

---

## DNF debuginfo-install Plugin

---

Install the associated debuginfo packages for a given package specification.

### Synopsis

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

### Arguments

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

### 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.)

### Examples

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





---

## DNF download Plugin

---

Download binary or source packages.

### Synopsis

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

### 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.

### Options

- `--help-cmd` Show this help.
- `--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.
- `--destdir` 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.

## 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 leaves Plugin

---

List installed packages not required by any other installed package.

### Synopsis

```
dnf leaves
```

### 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.

### 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.



### 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.

---

### 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 `repor_dir` parameter which sets path to local repository.

Other options and comments you can find in configuration file.





### Description

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

### Synopsis

```
dnf migrate [all|history|groups|yumdb]
```

### Commands

**dnf migrate [all]** Migrate all the data.

**dnf migrate history** Migrate only the history.

**dnf migrate groups** Migrate only the groups.

**dnf migrate yumdb** Migrate only the yumdb.



---

## DNF needs-restarting Plugin

---

Check for running processes that should be restarted.

### Synopsis

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

### 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.

### Options

`-u, --useronly`

Only consider processes belonging to the running user.



---

## DNF repoclosure Plugin

---

Display a list of unresolved dependencies for repositories.

### Synopsis

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

### 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.

### Options

- arch** <arch> Query only packages for specified architecture, can be specified multiple times (default is all architectures).
- 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> Check closure for this package only.
- repo** <repo id> Specify repo ids to query, can be specified multiple times (default is all enabled).

### 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
```

---

## DNF repograph Plugin

---

Output a full package dependency graph in dot format.

### Synopsis

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

### Description

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

### Options

Set what repos should be processed.

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

### 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
```



---

## DNF repomanage Plugin

---

Manage a directory of rpm packages.

### Synopsis

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

### Description

*repomanage* prints newest or oldest packages in specified directory for easy piping to xargs or similar programs.

### Options

Set what packages is displayed.

The following are mutually exclusive, i.e. only one can be specified. If no options specified - `--new` activated.

`--old` Show older packages.

`--new` Show newest packages.

### Optional 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.

## Examples

Display newest packages in current directory:

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

Display 2 newest packages in requires “home” directory:

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

Display older package separated by space in current directory:

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

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

### Synopsis

```
dnf reposync [-p <download-path>] ...
```

### Description

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

### Options

**-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.



## CHAPTER 17

---

### 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 versionlock Plugin

---

### Description

*versionlock* is a plugin that takes a set of names / versions for packages and excludes all other versions of those packages. This allows you to protect packages from being updated by newer versions.

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, 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 doing an 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. Versionlock plugin does not apply any excludes in non-transactional operations like *repoquery*, *list*, *info*, etc.

### Synopsis

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

### Arguments

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

### Subcommands

**dnf versionlock add <package-spec>** Add a versionlock for all available packages matching the spec. It means that only versions of packages represented by <package-spec> will be available for transaction operations.

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

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

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

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

## Configuration

/etc/dnf/plugins/versionlock.conf

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

**locklist** This option is a string with points to the file which will have 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-spec` (optionally prefixed with `!` for excludes). See `specifying_packages-label`.



## CHAPTER 19

---

### See Also

---

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



## CHAPTER 20

---

### Indices and tables

---

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