



Red Hat Enterprise Linux 10

Managing software with the DNF tool

Managing content in the RPM repositories by using the DNF software management tool

Red Hat Enterprise Linux 10 Managing software with the DNF tool

Managing content in the RPM repositories by using the DNF software management tool

Legal Notice

Copyright © Red Hat.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

Find, install, and utilize content distributed through the RPM repositories by using the DNF tool.

Table of Contents

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION	3
CHAPTER 1. DISTRIBUTION OF CONTENT IN RHEL 10	4
1.1. REPOSITORIES	4
1.2. APPLICATION STREAMS	4
CHAPTER 2. CONFIGURING DNF	6
2.1. VIEWING THE CURRENT DNF CONFIGURATIONS	6
2.2. SETTING DNF MAIN OPTIONS	6
2.3. ENABLING AND DISABLING DNF PLUGINS	6
2.4. EXCLUDING PACKAGES FROM DNF OPERATIONS	7
CHAPTER 3. SEARCHING FOR RHEL CONTENT	9
3.1. SEARCHING FOR SOFTWARE PACKAGES	9
3.2. LISTING SOFTWARE PACKAGES	9
3.3. DISPLAYING PACKAGE INFORMATION	10
3.4. LISTING PACKAGE GROUPS AND PACKAGES THEY PROVIDE	11
3.5. LISTING REPOSITORIES	11
3.6. SPECIFYING GLOB EXPRESSIONS IN DNF INPUT	12
3.7. ADDITIONAL RESOURCES	12
CHAPTER 4. INSTALLING RHEL CONTENT	13
4.1. INSTALLING PACKAGES	13
4.2. INSTALLING PACKAGE GROUPS	13
4.3. ADDITIONAL RESOURCES	14
CHAPTER 5. UPDATING RHEL CONTENT	15
5.1. CHECKING FOR UPDATES	15
5.2. UPDATING PACKAGES	15
5.3. UPDATING SECURITY-RELATED PACKAGES	16
CHAPTER 6. AUTOMATING SOFTWARE UPDATES IN RHEL	17
6.1. INSTALLING DNF AUTOMATIC	17
6.2. DNF AUTOMATIC CONFIGURATION FILE	17
6.3. ENABLING DNF AUTOMATIC	18
6.4. OVERVIEW OF THE SYSTEMD TIMER UNITS INCLUDED IN THE DNF-AUTOMATIC PACKAGE	19
CHAPTER 7. REMOVING RHEL CONTENT	21
7.1. REMOVING INSTALLED PACKAGES	21
7.2. REMOVING PACKAGE GROUPS	21
7.3. ADDITIONAL RESOURCES	21
CHAPTER 8. HANDLING PACKAGE MANAGEMENT HISTORY	22
8.1. LISTING DNF TRANSACTIONS	22
8.2. REVERTING DNF TRANSACTIONS	23
8.2.1. Reverting a single DNF transaction	23
8.2.2. Reverting multiple DNF transactions	24
CHAPTER 9. MANAGING CUSTOM SOFTWARE REPOSITORIES	26
APPENDIX A. DNF COMMANDS LIST	28
A.1. COMMANDS FOR LISTING CONTENT IN RHEL	28
A.2. COMMANDS FOR INSTALLING CONTENT IN RHEL	29
A.3. COMMANDS FOR REMOVING CONTENT IN RHEL	29

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your feedback on our documentation. Let us know how we can improve it.

Submitting feedback through Jira (account required)

1. Log in to the [Jira](#) website.
2. Click **Create** in the top navigation bar.
3. Enter a descriptive title in the **Summary** field.
4. Enter your suggestion for improvement in the **Description** field. Include links to the relevant parts of the documentation.
5. Click **Create** at the bottom of the dialogue.

CHAPTER 1. DISTRIBUTION OF CONTENT IN RHEL 10

In Red Hat Enterprise Linux (RHEL) 10, software is distributed through different repositories. You can access applications and core components through these repositories to ensure your system remains stable and up-to-date.

1.1. REPOSITORIES

Red Hat Enterprise Linux (RHEL) 10 repositories, such as BaseOS or AppStream, provide the specific software, operating system functionality, and various system components you need for a stable and supported environment.

Red Hat Enterprise Linux distributes content through different repositories, for example, BaseOS, AppStream, CodeReady Linux Builder, and Supplementary. In addition, specific content is available in the Red Hat Enterprise Linux Add-on repositories, such as High Availability.

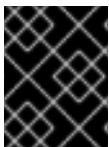
The main repositories that RHEL uses to distribute its content include the following repositories:

BaseOS

Content in the BaseOS repository consists of the core set of the underlying operating system functionality that provides the foundation for all installations. This content is available in the RPM format and is subject to support terms similar to those in earlier releases of RHEL.

AppStream

Content in the AppStream repository includes additional user-space applications, runtime languages, and databases in support of the varied workloads and use cases.



IMPORTANT

Both the BaseOS and AppStream content sets are required by RHEL and are available in all RHEL subscriptions.

CodeReady Linux Builder

The CodeReady Linux Builder repository is available with all RHEL subscriptions. It provides additional packages for use by developers. Red Hat does not support packages included in the CodeReady Linux Builder repository.

Additional resources

- [Package manifest](#)

1.2. APPLICATION STREAMS

Red Hat provides multiple versions of user-space components as Application Streams, and they are updated more frequently than the core operating system packages. This provides more flexibility to customize RHEL without impacting the underlying stability of the platform or specific deployments.

Application Streams are available in the following formats:

- RPM format
- Software Collections

RHEL 10 provides initial Application Stream versions as RPMs, which you can install by using the **dnf install** command.



IMPORTANT

Each Application Stream has its own life cycle, and it can be the same or shorter than the life cycle of Red Hat Enterprise Linux 10. See [Red Hat Enterprise Linux Application Streams Life Cycle](#).

Always determine which version of an Application Stream you want to install, and make sure to review the Red Hat Enterprise Linux Application Stream life cycle first.

Additional resources

- [Red Hat Enterprise Linux 10: Application Compatibility Guide](#)
- [Package manifest](#)
- [Red Hat Enterprise Linux Application Streams Life Cycle](#)

CHAPTER 2. CONFIGURING DNF

Customize software management on your Red Hat Enterprise Linux (RHEL) 10 system by configuring global settings and repository options for the DNF tool. You can adjust these parameters to optimize package downloads and ensure your environment meets specific operational requirements.

The configuration of DNF and related utilities is stored in the **[main]** section of the `/etc/dnf/dnf.conf` file.

2.1. VIEWING THE CURRENT DNF CONFIGURATIONS

Verify your software management settings by viewing the active DNF configuration. Reviewing these parameters ensures that your system uses the correct global and repository settings for efficient package management.

The **[main]** section in the `/etc/dnf/dnf.conf` file contains only the settings that have been explicitly set. However, you can display all settings of the **[main]** section, including the ones that have not been set and which, therefore, use their default values.

Procedure

- Display the global DNF configuration:

```
# dnf config-manager --dump
```

2.2. SETTING DNF MAIN OPTIONS

To control how DNF operates, configure key-value pairs in the **[main]** section of the `/etc/dnf/dnf.conf` file.

Procedure

1. Edit the `/etc/dnf/dnf.conf` file.
2. Update the **[main]** section according to your requirements.
3. Save the changes.

2.3. ENABLING AND DISABLING DNF PLUGINS

Extend the functionality of the DNF tool by managing its plugins. By enabling or disabling plugins, you can activate or remove specific features to align with your environment's operational requirements.

In the DNF tool, plugins are loaded by default. However, you can influence which plugins DNF loads.

Every installed plugin can have its own configuration file in the `/etc/dnf/plugins/` directory. Name plugin configuration files in the `<plugin_name>.conf` directory. By default, plugins are typically enabled. You can manage plugins by either using different **dnf** commands or modifying the **[main]** section of the plugin's configuration file.



WARNING

Disable all plugins only for diagnosing a potential problem. DNF requires certain plugins, such as **product-id** and **subscription-manager**, and disabling them causes Red Hat Enterprise Linux to not be able to install or update software from the Content Delivery Network (CDN).

Procedure

- Use one of the following methods to influence how DNF uses plugins:
 - To enable or disable loading of DNF plugins globally, add the **plugins** parameter to the **[main]** section of the `/etc/dnf/dnf.conf` file.
 - Set **plugins=1** (default) to enable loading of all DNF plugins.
 - Set **plugins=0** to disable loading of all DNF plugins.

- To disable a particular plugin, add **enabled=False** to the **[main]** section in the `/etc/dnf/plugins/<plugin_name>.conf` file:

```
[main]
enabled=False
```

- To disable all DNF plugins for a particular command, append the **--noplugins** option to the command. For example, to disable DNF plugins for a single update command, enter:

```
# dnf --noplugins update
```

- To disable certain DNF plugins for a single command, append the **--disableplugin=<plugin_name>** option to the command. For example, to disable a certain DNF plugin for a single update command, enter:

```
# dnf update --disableplugin=<plugin_name>
```

- To enable certain DNF plugins for a single command, append the **--enableplugin=<plugin_name>** option to the command. For example, to enable a certain DNF plugin for a single update command, enter:

```
# dnf update --enableplugin=<plugin_name>
```

2.4. EXCLUDING PACKAGES FROM DNF OPERATIONS

To prevent specific software from being installed or updated, exclude packages from DNF operations by using the **excludepkgs** option.

You can configure DNF to exclude packages from any DNF operation by using the **excludepkgs** option. You can define the **excludepkgs** option in the **[main]** or the repository section of the `/etc/dnf/dnf.conf` DNF configuration file.

**NOTE**

You can temporarily disable excluding the configured packages from an operation by using the **--disableexcludes** option.

Procedure

- Exclude packages from the DNF operation by adding the following line to the **/etc/dnf/dnf.conf** file:

```
excludepkgs=<package_name_1>,<package_name_2> ...
```

Additional resources

- [Specifying global expressions in DNF input](#)

CHAPTER 3. SEARCHING FOR RHEL CONTENT

Search and examine content in the Red Hat Enterprise Linux (RHEL) 10 AppStream and BaseOS repositories by using the DNF tool. For example, you can identify specific packages or package groups to ensure you install the exact components required for your environment's stability and security.

3.1. SEARCHING FOR SOFTWARE PACKAGES

Identify and locate the specific software for your Red Hat Enterprise Linux (RHEL) 10 system by searching for packages with the DNF tool. You can search by name or summary to ensure you find and install the correct components required for your environment.

Procedure

- Depending on your scenario, use one of the following options to search the repository:
 - To search for a term in the name or summary of packages, enter:

```
$ dnf search <term>
```

- To search for a term in the name, summary, or description of packages, enter:

```
$ dnf search --all <term>
```

Note that searching additionally in the description by using the **--all** option is slower than a normal search operation.

- To search for a package name and list the package name and its version in the output, enter:

```
$ dnf repoquery <package_name>
```

- To search for which package provides a file, specify the file name or the path to the file:

```
$ dnf provides <file_name>
```

3.2. LISTING SOFTWARE PACKAGES

List available or installed packages on your Red Hat Enterprise Linux (RHEL) 10 system by using the DNF tool. You can use this information to confirm current versions and identify ready-to-install content to ensure your environment stays consistent and up-to-date.

Procedure

- List the latest versions of all available packages, including architectures, version numbers, and the repository they were installed from:

```
$ dnf list --all
```

```
...
```

```
postgresql.x86_64      16.4-1.el10    rhel-AppStream
postgresql-contrib.x86_64 16.4-1.el10    rhel-AppStream
```

```
postgresql-docs.x86_64      16.4-1.el10    rhel-AppStream
postgresql-jdbc.noarch     42.7.1-6.el10  rhel-AppStream
...
```

The @ sign in front of a repository indicates that the package in this line is currently installed.

Alternatively, to display all available packages, including version numbers and architectures, enter:

```
$ dnf repoquery

...
postgresql-0:16.4-1.el10.x86_64
postgresql-contrib-0:16.4-1.el10.x86_64
postgresql-docs-0:16.4-1.el10.x86_64
postgresql-jdbc-0:42.7.1-6.el10.noarch
postgresql-odbc-0:16.00.0000-4.el10.x86_64
...
```

Optionally, you can filter the output by using other options instead of **--all**, for example:

- Use **--installed** to list only installed packages.
- Use **--available** to list all available packages.
- Use **--upgrades** to list packages for which newer versions are available.

Additional resources

- [Specifying global expressions in DNF input](#)

3.3. DISPLAYING PACKAGE INFORMATION

View detailed metadata for specific software in Red Hat Enterprise Linux (RHEL) 10 by displaying package information with the DNF tool. You can use this information to verify that a package meets your system requirements and security standards before installation.

You can display the following types of information related to the package:

- Version
- Release
- Architecture
- Package size
- Description

Procedure

- Display information about one or more available packages:

```
$ dnf info <package_name>
```

This command displays the information for the currently installed package and, if available, its newer versions that are in the repository. Alternatively, use the following command to display the information for all packages with the specified name in the repository:

```
$ dnf repoquery --info <package_name>
```

Additional resources

- [Specifying global expressions in DNF input](#)

3.4. LISTING PACKAGE GROUPS AND PACKAGES THEY PROVIDE

List available package groups and their contents for your Red Hat Enterprise Linux (RHEL) 10 system by using the DNF tool. You can list package groups to quickly verify and install all required tools for specialized environments.

Procedure

1. List both installed and available groups:

```
$ dnf group list
```

Note that you can filter the results by appending the **--installed** and **--available** option to the **dnf group list** command. By using the **--hidden** option, you can display hidden groups in the output.

2. List mandatory, optional, and default packages contained in a particular group:

```
$ dnf group info "<group_name>"
```

3. Optional: View the number of installed and available groups:

```
$ dnf group summary
```

Additional resources

- [Specifying global expressions in DNF input](#)

3.5. LISTING REPOSITORIES

List enabled and disabled repositories on your Red Hat Enterprise Linux (RHEL) 10 system by using the DNF tool. You can review the repositories to identify available content and troubleshoot repository availability to ensure your environment can access the necessary software updates.

Procedure

1. List all enabled repositories on your system:

```
$ dnf repolist
```

To display only certain repositories, append one of the following options to the command:

- Append **--disabled** to list only disabled repositories.
 - Append **--all** to list both enabled and disabled repositories.
2. Optional: List additional information about the repositories:

```
$ dnf repoinfo <repository_name>
```

Additional resources

- [Specifying global expressions in DNF input](#)

3.6. SPECIFYING GLOB EXPRESSIONS IN DNF INPUT

Write more expressive DNF commands on your Red Hat Enterprise Linux (RHEL) 10 system by appending one or more glob expressions as arguments.

Many DNF commands accept glob expressions in place of package names, file paths, and other parameters. Glob expressions are strings of characters that contain one or more of the wildcard characters. By using glob expressions, you can efficiently search for large groups of related software and match paths without requiring exact matches for every item.

Procedure

- Use one of the following methods if you use global expressions in **dnf** commands:
 - Enclose the entire global expression in single or double quotation marks:

```
# dnf provides "*/<file_name>"
```

Note that you must precede **<file_name>** either by **/** for an absolute path or ***/** to use a wildcard if the full path is unknown.

- Escape the wildcard characters by preceding them with a backslash (****) character:

```
# dnf provides \*/<file_name>
```

3.7. ADDITIONAL RESOURCES

- [Commands for listing content in RHEL](#)

CHAPTER 4. INSTALLING RHEL CONTENT

Manually install software packages and package groups, which are not part of the default installation, on your Red Hat Enterprise Linux (RHEL) 10 system by using the DNF tool.

4.1. INSTALLING PACKAGES

Manually install software which is not part of the default installation on your Red Hat Enterprise Linux 10 (RHEL) system by using the DNF tool. During the package installation, DNF automatically resolves and installs package dependencies.

During the package installation, DNF automatically resolves and installs package dependencies.

Procedure

- Use one of the following methods to install packages:

- To install packages from the repositories, enter:

```
# dnf install <package_name_1> <package_name_2> ...
```

If you install packages on a system that supports multiple architectures, such as **i686** and **x86_64**, you can specify the architecture of the package by appending it to the package name:

```
# dnf install <package_name>.<architecture>
```

- To install a package if you only know the path to the file the package provides but not the package name, you can use this path to install the corresponding package:

```
# dnf install <path_to_file>
```

- To install a local RPM file, enter:

```
# dnf install <path_to_RPM_file>
```

If the package has dependencies, specify the paths to these RPM files as well. Otherwise, DNF downloads the dependencies from the repositories or fails if they are not available in the repositories.

Additional resources

- [Searching for RHEL content](#)

4.2. INSTALLING PACKAGE GROUPS

Package groups bundle multiple packages. Install all packages assigned to a package group to your Red Hat Enterprise Linux (RHEL) 10 system in a single step by using the DNF tool.

Prerequisites

- You know the name or ID of the group you want to install. For more information, see [Listing package groups and packages they provide](#).

Procedure

- Install a package group:

```
# dnf group install <group_name_or_ID>
```

4.3. ADDITIONAL RESOURCES

- [Commands for installing content in RHEL](#)

CHAPTER 5. UPDATING RHEL CONTENT

To maintain the security and performance of your Red Hat Enterprise Linux (RHEL) 10 system, apply software updates by using the DNF tool. Regularly installing package and security updates ensures your environment remains stable, up-to-date, and protected against known vulnerabilities.

With the DNF tool, you can list packages that need updating and choose to update a single package, multiple packages, or all packages at once.

TIP

You can manage software updates in the [Red Hat Enterprise Linux web console](#), which provides a graphical interface for DNF.

5.1. CHECKING FOR UPDATES

To ensure your Red Hat Enterprise Linux (RHEL) 10 system remains secure and efficient, check for available software updates by using the DNF tool. By identifying pending updates, you can plan maintenance windows and prioritize security fixes before applying changes to your environment.

Procedure

- Check the available updates for installed packages:

```
# dnf check-update
```

The output returns the list of packages and their dependencies that have an update available.

5.2. UPDATING PACKAGES

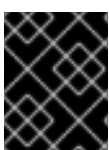
To maintain the security and stability of your Red Hat Enterprise Linux (RHEL) 10 system, install the latest software updates by using the DNF tool. Regularly applying package updates ensures your environment has essential security patches and bug fixes.

You can use DNF to update a single package, a package group, or all packages and their dependencies at once. If any of the packages you want to update have dependencies, DNF updates these dependencies as well.



IMPORTANT

When applying updates to the kernel, **dnf** always installs a new kernel regardless of whether you are using the **dnf upgrade** or **dnf install** command. Note that this only applies to packages identified by using the **installonlypkgs** DNF configuration option. Such packages include the **kernel**, **kernel-core**, and **kernel-modules** packages.



IMPORTANT

If you upgraded the GRUB boot loader packages on a BIOS or IBM Power system, reinstall GRUB. See [Reinstalling GRUB](#).

Procedure

- Depending on your scenario, use one of the following options to apply updates:

- To update all packages and their dependencies, enter:

```
# dnf upgrade
```

- To update a single package, enter:

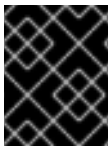
```
# dnf upgrade <package_name>
```

- To update packages only from a specific package group, enter:

```
# dnf group upgrade <group_name>
```

5.3. UPDATING SECURITY-RELATED PACKAGES

Protect your Red Hat Enterprise Linux (RHEL) 10 system against known and newly discovered threats and vulnerabilities by installing security-related updates with the DNF tool.



IMPORTANT

If you upgraded the GRUB boot loader packages on a BIOS or IBM Power system, reinstall GRUB. See [Reinstalling GRUB](#).

Procedure

- Depending on your scenario, use one of the following options to apply updates:
 - To upgrade to the latest available packages that have security errata, enter:

```
# dnf upgrade --security
```

- To upgrade to the last security errata packages, enter:

```
# dnf upgrade-minimal --security
```

Additional resources

- [Managing and monitoring security updates](#)

CHAPTER 6. AUTOMATING SOFTWARE UPDATES IN RHEL

Ensure your Red Hat Enterprise Linux (RHEL) 10 system remains secure and up-to-date by automating software updates with the DNF Automatic tool. Automatically installing security patches and bug fixes helps you maintain environment stability while reducing the effort for manual system maintenance.

DNF Automatic is an alternative command-line interface to DNF that is suited for automatic and regular execution by using **systemd** timers, cron jobs, and other such tools.

DNF Automatic synchronizes package metadata as needed, checks for updates available, and then performs one of the following actions depending on how you configure the tool:

- Exit
- Download updated packages
- Download and apply the updates

The outcome of the operation is then reported by a selected mechanism, such as the standard output or email.

6.1. INSTALLING DNF AUTOMATIC

Enable automatic software updates on your Red Hat Enterprise Linux (RHEL) 10 system by installing the DNF Automatic tool. Using this tool ensures your environment remains secure and up-to-date with the latest patches while significantly reducing the time and effort required for manual system maintenance.

Procedure

- Install the **dnf-automatic** package:

```
# dnf install dnf-automatic
```

Verification

- Verify the successful installation by confirming the presence of the **dnf-automatic** package:

```
# rpm -qi dnf-automatic
```

6.2. DNF AUTOMATIC CONFIGURATION FILE

The DNF Automatic configuration file defines the parameters for automated software updates in Red Hat Enterprise Linux (RHEL) 10. By managing these settings, you can customize update behaviors to maintain system security consistently with minimal manual effort.

By default, DNF Automatic uses **/etc/dnf/automatic.conf** as its configuration file to define its behavior. The configuration file has the following topical sections:

- **[commands]**
Sets the mode of operation of DNF Automatic.

**WARNING**

Settings of the operation mode from the **[commands]** section are overridden by settings used by a **systemd** timer unit for all timer units except **dnf-automatic.timer**.

- **[emitters]**
Defines how the results of DNF Automatic are reported.
- **[command]**
Defines the command emitter configuration.
- **[command_email]**
Provides the email emitter configuration for an external command used to send email.
- **[email]**
Provides the email emitter configuration.
- **[base]**
Overrides settings from the main configuration file of DNF.

With the default settings of the **/etc/dnf/automatic.conf** file, DNF Automatic checks for available updates, downloads them, and reports the results to standard output.

Additional resources

- [Overview of the systemd timer units included in the dnf-automatic package](#)

6.3. ENABLING DNF AUTOMATIC

Enable automated software updates on Red Hat Enterprise Linux (RHEL) 10 by activating DNF Automatic timer units. By enabling these units, you ensure your system remains consistently protected and up-to-date without requiring manual effort.

To run DNF Automatic once, you must start a **systemd** timer unit. However, if you want to run DNF Automatic periodically, you must enable a timer unit.

You can use one of the timer units provided in the **dnf-automatic** package, or you can create a drop-in file for the timer unit to adjust the execution time.

You can use the following timers:

- **dnf-automatic-download.timer**: Downloads available updates.
- **dnf-automatic-install.timer**: Downloads and installs available updates.
- **dnf-automatic-notifyonly.timer**: Reports available updates.
- **dnf-automatic.timer**: Downloads, downloads and installs, or reports available updates.

Prerequisites

- You specified the behavior of DNF Automatic by modifying the `/etc/dnf/automatic.conf` configuration file.

Procedure

- To enable and execute a **systemd** timer unit immediately, enter:

```
# systemctl enable --now <timer_name>
```

If you want to only enable the timer without executing it immediately, omit the **--now** option.

Verification

- Verify that the timer is enabled:

```
# systemctl status <systemd_timer_unit>
```

- Optional: Check when each of the timers on your system ran the last time:

```
# systemctl list-timers --all
```

Additional resources

- [Overview of the systemd timer units included in the dnf-automatic package](#)

6.4. OVERVIEW OF THE SYSTEMD TIMER UNITS INCLUDED IN THE DNF-AUTOMATIC PACKAGE

The **systemd** timer units in the **dnf-automatic** package provide the scheduling framework for automated software updates in Red Hat Enterprise Linux (RHEL) 10. Understanding these units helps you to customize the timing and frequency of update checks and patch applications.

The **systemd** timer units take precedence and override the settings in the `/etc/dnf/automatic.conf` configuration file when downloading and applying updates. For example if you set **download_updates = yes** in the `/etc/dnf/automatic.conf` configuration file, but you have activated the **dnf-automatic-notifyonly.timer** unit, DNF will not download the packages.

The **dnf-automatic** package provides following **systemd** timer units:

Table 6.1. systemd timers distributed with dnf-automatic

Timer unit	Function	Overrides the <code>apply_updates</code> and <code>download_updates</code> settings in the <code>[commands]</code> section of the <code>/etc/dnf/automatic.conf</code> file?

Timer unit	Function	Overrides the apply_updates and download_updates settings in the [commands] section of the /etc/dnf/automatic.conf file?
dnf-automatic-download.timer	<p>Downloads packages to cache and makes them available for updating.</p> <p>This timer unit does not install the updated packages. To perform the installation, you must run the dnf update command.</p>	Yes
dnf-automatic-install.timer	<p>Downloads and installs updated packages.</p>	Yes
dnf-automatic-notifyonly.timer	<p>Downloads only repository data to keep the repository cache up-to-date and notifies you about available updates.</p> <p>This timer unit does not download or install the updated packages.</p>	Yes
dnf-automatic.timer	<p>The behavior of this timer when downloading and applying updates is specified by the settings in the /etc/dnf/automatic.conf configuration file.</p> <p>This timer downloads packages, but does not install them.</p>	No

CHAPTER 7. REMOVING RHEL CONTENT

To optimize your environment, remove unnecessary software from Red Hat Enterprise Linux (RHEL) 10 by using the DNF tool. Uninstalling unused packages and functional groups ensures your system remains clean, efficient, and secure.

7.1. REMOVING INSTALLED PACKAGES

To optimize your Red Hat Enterprise Linux (RHEL) 10 system, remove unnecessary software packages by using the DNF tool. Uninstalling unused software ensures your environment remains secure and streamlined.

You can use DNF to remove a single package or multiple packages installed on your system. If any of the packages you want to remove have unused dependencies, DNF uninstalls these dependencies as well.

Procedure

- Remove particular packages:

```
# dnf remove <package_name_1> <package_name_2> ...
```

7.2. REMOVING PACKAGE GROUPS

To optimize your Red Hat Enterprise Linux (RHEL) 10 system, uninstall collections of related software by removing package groups with the DNF tool. Uninstalling entire groups ensures your environment remains efficient and secure by removing all unnecessary components at once.

Package groups bundle multiple packages. You can use package groups to remove all packages assigned to a group in a single step.

Prerequisites

- You know the name or ID of the group you want to remove. For more information, see [Listing package groups and packages they provide](#).

Procedure

- Remove package groups by the group name or group ID:

```
# dnf group remove <group_name> <group_ID>
```

7.3. ADDITIONAL RESOURCES

- [Commands for removing content in RHEL](#)

CHAPTER 8. HANDLING PACKAGE MANAGEMENT HISTORY

Track and manage software changes on your Red Hat Enterprise Linux (RHEL) 10 systems by using the DNF history database. By reviewing or reverting past transactions, you can quickly recover from accidental configuration changes.

You can perform various operations on the package management history by using the **dnf history** command. You can review the following information related to the DNF transactions:

- Timeline of transactions.
- Dates and times the transactions occurred.
- Number of packages affected by the transactions.
- Whether the transactions succeeded or were aborted.
- If the RPM database was changed between the transactions.

You can also use the **dnf history** command to undo operations performed during the transaction.

8.1. LISTING DNF TRANSACTIONS

To audit software changes or identify specific package operations, list the package management history on Red Hat Enterprise Linux (RHEL) 10 by using the DNF tool. Reviewing these transactions ensures you have a clear record of system modifications for maintenance and troubleshooting purposes.

By using the DNF tool, you can perform the following tasks:

- List the latest transactions.
- List the latest operations for a selected package.
- Display details of a particular transaction.

Procedure

- Depending on your scenario, use one of the following options to display transaction information:
 - To display a list of all the latest DNF transactions, enter:

```
# dnf history
```

The output contains the following information:

- The **Action(s)** column displays which type of action was performed during a transaction, for example, Install (**I**), Upgrade (**U**), Remove (**E**), and other actions.
- The **Altered** column displays the number of actions performed during the transaction. The number of actions can also be followed by the result of the transaction. For more information about the values of the **Action(s)** and **Altered** columns, see the **dnf(8)** man page.
- To display a list of all the latest operations for a selected package, enter:

```
# dnf history list <package_name>
```

- To display details of a particular transaction, enter:

```
# dnf history info <transaction_ID>
```

Additional resources

- [Specifying global expressions in DNF input](#)

8.2. REVERTING DNF TRANSACTIONS

To undo operations performed during DNF transactions on Red Hat Enterprise Linux (RHEL) 10, revert these transactions by using the DNF tool. By reverting transactions, you can quickly restore your system to a previous state. For example, if you installed several packages by using the **dnf install** command, you can uninstall these packages at once by reverting the installation transaction.

You can revert DNF transactions the following ways:

- Revert a single DNF transaction by using the **dnf history undo** command.
- Revert all DNF transactions performed between the specified transaction and the last transaction by using the **dnf history rollback** command.



IMPORTANT

Downgrading RHEL system packages to an older version by using the **dnf history undo** and **dnf history rollback** command is not supported. This concerns especially the **selinux**, **selinux-policy-***, **kernel**, and **glibc** packages, and dependencies of **glibc** such as **gcc**. Therefore, downgrading a system to a minor version (for example, from RHEL 10.1 to RHEL 10.0) is not recommended because it might leave the system in an incorrect state.

8.2.1. Reverting a single DNF transaction

To undo operations performed during a single DNF transaction on Red Hat Enterprise Linux (RHEL) 10, revert this transaction by using the DNF tool. By reverting the environment to a specific point in your package management history, you can recover from accidental changes and errors. Reverting a single transaction in DNF history will not undo or modify more recent transactions.

You can revert the transaction's steps by using the **dnf history undo** command:

- If the transaction installed a new package, **dnf history undo** uninstalls the package.
- If the transaction uninstalled a package, **dnf history undo** reinstalls the package.
- The **dnf history undo** command also attempts to downgrade all updated packages to their previous versions if the older packages are still available.



NOTE

If an older package version is not available, the downgrade by using the **dnf history undo** command fails.

Procedure

1. Identify the ID of a transaction you want to revert:

```
# dnf history
```

ID	Command line	Date and time	Action(s)	Altered
13	install zip	2022-11-03 10:49	Install	1
12	install unzip	2022-11-03 10:49	Install	1

2. Optional: Verify that this is the transaction you want to revert by displaying its details:

```
# dnf history info <transaction_ID>
```

3. Revert the transaction:

```
# dnf history undo <transaction_ID>
```

For example, if you want to uninstall the previously installed **unzip** package, enter:

```
# dnf history undo 12
```

8.2.2. Reverting multiple DNF transactions

To undo operations performed during DNF transactions on Red Hat Enterprise Linux (RHEL) 10, revert these transactions by using the DNF tool. By reverting the environment to a specific point in your package management history, you can recover from accidental changes and errors.

You can revert all DNF transactions performed between a specified transaction and the last transaction by using the **dnf history rollback** command. Note that the transaction specified by the transaction ID remains unchanged.

Procedure

1. Identify the transaction ID of the state you want to revert to:

```
# dnf history
```

ID	Command line	Date and time	Action(s)	Altered
14	install wget	2022-11-03 10:49	Install	1
13	install unzip	2022-11-03 10:49	Install	1
12	install vim-X11	2022-11-03 10:20	Install	171 EE

2. Revert specified transactions:

```
# dnf history rollback <transaction_ID>
```

For example, to revert to the state before the **wget** and **unzip** packages were installed, enter:

```
# dnf history rollback 12
```

Alternatively, to revert all transactions in the transaction history, use the transaction ID **1**:

```
| # dnf history rollback 1
```

CHAPTER 9. MANAGING CUSTOM SOFTWARE REPOSITORIES

To provide your Red Hat Enterprise Linux (RHEL) 10 systems with access to organization-specific or third-party software, add and configure custom repositories by using the DNF tool. By managing these custom sources, you can install and update specialized packages required for your environment.

You can configure a repository in the `/etc/dnf/dnf.conf` file or in a `.repo` file in the `/etc/yum.repos.d/` directory.

The `/etc/dnf/dnf.conf` file contains the **[main]** section and can contain one or more repository sections with a unique repository ID in brackets (`[]`), for example, (`[<repository-ID>]`). You can use these sections to define individual DNF repositories by setting repository-specific options. Note that repository IDs must be unique. The values you define in individual repository sections of the `/etc/dnf/dnf.conf` file override values set in the **[main]** section for this repository.

For a complete list of available repository ID options, see the **[<repository_ID>] OPTIONS** section of the **dnf.conf(5)** man page.

Consider adding your custom repositories in separate `.repo` files instead of the `/etc/dnf/dnf.conf` DNF configuration file to avoid possible issues if other programs modify the DNF configuration file.

You can add the DNF repository to your system by using the **dnf config-manager --add-repo** command. Repositories that you add with this command are enabled by default. However, you can also use the **dnf config-manager** command to disable the repository.



WARNING

Obtaining and installing software packages from unverified or untrusted sources other than Red Hat certificate-based **Content Delivery Network** (CDN) is a potential security risk, and can lead to security, stability, compatibility, and maintainability issues.

Procedure

1. Add a repository to your system:

```
# dnf config-manager --add-repo <repository_URL>
```

2. Review and, optionally, update the repository settings that the previous command created in the `/etc/yum.repos.d/<repository_URL>.repo` file:

```
# cat /etc/yum.repos.d/<repository_URL>.repo
```

3. Optional: Disable the DNF repository added to your system:

```
# dnf config-manager --disable <repository_ID>
```

To re-enable the repository, enter:

```
# dnf config-manager --enable <repository_ID>
```

APPENDIX A. DNF COMMANDS LIST

Use the following DNF commands to install, update, remove, and query system content.

A.1. COMMANDS FOR LISTING CONTENT IN RHEL

Use the following DNF commands to search and examine available or installed software for your Red Hat Enterprise Linux (RHEL) 10 system.

Command	Description
dnf search <term>	Search for a package by using term related to the package.
dnf repoquery <package_name>	Search for enabled DNF repositories for a selected package and its version.
dnf list	List information about all installed and available packages.
dnf list --installed dnf repoquery --installed	List all packages installed on your system.
dnf list --available dnf repoquery	List all packages in all enabled repositories that are available to install.
dnf repolist	List all enabled repositories on your system.
dnf repolist --disabled	List all disabled repositories on your system.
dnf repolist --all	List both enabled and disabled repositories.
dnf repoinfo	List additional information about the repositories.
dnf info <package_name> dnf repoquery --info <package_name>	Display details of an available package.
dnf repoquery --info --installed <package_name>	Display details of a package installed on your system.
dnf group summary	View the number of installed and available groups.
dnf group list	List all installed and available groups.

Command	Description
dnf group info <group_name>	List mandatory and optional packages included in a particular group.

A.2. COMMANDS FOR INSTALLING CONTENT IN RHEL

Use the following commands to install software on your Red Hat Enterprise Linux (RHEL) 10 system.

Command	Description
dnf install <package_name>	Install a package.
dnf install <package_name_1> <package_name_2>	Install multiple packages and their dependencies simultaneously.
dnf install <package_name>.<architecture>	Specify the architecture of the package by appending it to the package name when installing packages on a <i>multilib</i> system (AMD64, Intel 64 machine).
dnf install <path_to_file>	Install a binary by using the path to the binary as an argument.
dnf install <path_to_RPM_file>	Install a local RPM file.
dnf install <package_url>	Install a remote package by using a package URL.
dnf group install <group_name>	Install a package group by a group name.
dnf group install <group_ID>	Install a package group by the group ID.

A.3. COMMANDS FOR REMOVING CONTENT IN RHEL

Use the following DNF commands to uninstall software from your Red Hat Enterprise Linux (RHEL) 10 system.

Command	Description
dnf remove <package_name>	Remove a particular package and all dependent packages.
dnf remove <package_name_1> <package_name_2>	Remove multiple packages and their unused dependencies simultaneously.
dnf group remove <group_name>	Remove a package group by the group name.

Command	Description
dnf group remove <group_ID>	Remove a package group by the group ID.