



# Red Hat Enterprise Linux 9

## 使用 DNF 工具管理软件

在 Red Hat Enterprise Linux 9 中使用 DNF 管理软件的指南



# Red Hat Enterprise Linux 9 使用 DNF 工具管理软件

---

在 Red Hat Enterprise Linux 9 中使用 DNF 管理软件的指南

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

## 法律通告

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Managing\_software\_with\_the\_DNF\_tool.ent file |.

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

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

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

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> 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.

## 摘要

本文档论述了使用 Red Hat Enterprise Linux 9 中的 DNF 工具搜索、发现、安装和使用 AppStream 和 BaseOS 软件仓库中的内容。这包括如何使用模块、应用程序流和配置集的描述。

## 目录

使开源包含更多 .....	4
对红帽文档提供反馈 .....	5
第 1 章 RED HAT ENTERPRISE LINUX 9 中的软件管理工具 .....	6
第 2 章 RHEL 9 发布的内容 .....	7
2.1. 软件仓库	7
2.2. 应用程序流	7
2.3. 模块	8
2.4. 模块流	8
2.5. 模块配置集	8
第 3 章 配置 DNF .....	10
3.1. 查看当前的 DNF 配置	10
3.2. 设置 DNF 主选项	10
3.3. 使用 DNF 插件	10
3.3.1. 管理 DNF 插件	10
3.3.2. 启用和禁用 DNF 插件	10
第 4 章 搜索 RHEL 9 内容 .....	12
4.1. 搜索软件包	12
4.2. 列出软件包	12
4.3. 列出软件仓库	13
4.4. 显示软件包信息	13
4.5. 列出软件包组	14
4.6. 列出可用模块	14
4.7. 在 DNF 输入中指定全局表达式	15
4.8. 其它资源	15
第 5 章 安装 RHEL 9 内容 .....	16
5.1. 安装软件包	16
5.2. 安装软件包组	16
5.3. 运行安装的内容	17
5.4. 其它资源	17
第 6 章 更新 RHEL 9 内容 .....	18
6.1. 检查更新	18
6.2. 更新软件包	18
6.3. 更新与安全相关的软件包	19
第 7 章 在 RHEL 9 中自动化软件更新 .....	20
7.1. 安装 DNF AUTOMATIC	20
7.2. DNF AUTOMATIC 配置文件	20
7.3. 启用 DNF AUTOMATIC	21
7.4. DNF-AUTOMATIC 软件包中包含的 SYSTEMD 计时器单元概述	22
第 8 章 删除 RHEL 9 内容 .....	24
8.1. 删除安装的软件包	24
8.2. 删除软件包组	24
8.3. 其它资源	24
第 9 章 处理软件包管理历史记录 .....	26
9.1. 列出事务	26

9.2. 恢复事务	26
9.3. 重复事务	27
<b>第 10 章 管理自定义软件存储库</b> .....	<b>28</b>
10.1. 设置 DNF 软件仓库选项	28
10.2. 添加 DNF 软件仓库	28
10.3. 启用 DNF 软件仓库	29
10.4. 禁用 DNF 软件仓库	29
<b>附录 A. DNF 命令列表</b> .....	<b>30</b>
A.1. 在 RHEL 9 中列出内容的命令	30
A.2. 在 RHEL 9 中安装内容的命令	31
A.3. 在 RHEL 9 中删除内容的命令	31



## 使开源包含更多

红帽致力于替换我们的代码、文档和 Web 属性中存在问题的语言。我们从这四个术语开始：master、slave、黑名单和白名单。由于此项工作十分艰巨，这些更改将在即将推出的几个发行版本中逐步实施。详情请查看 [CTO Chris Wright 的信息](#)。



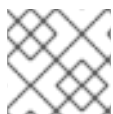
## 对红帽文档提供反馈

我们感谢您对文档提供反馈信息。请让我们了解如何改进文档。

- 关于特定内容的简单评论：
  1. 请确定您使用 *Multi-page HTML* 格式查看文档。另外，确定 **Feedback** 按钮出现在文档页的右上方。
  2. 用鼠标指针高亮显示您想评论的文本部分。
  3. 点在高亮文本上弹出的 **Add Feedback**。
  4. 按照显示的步骤操作。
- 要通过 Bugzilla 提交反馈，请创建一个新的 ticket：
  1. 进入 [Bugzilla](#) 网站。
  2. 在 Component 中选择 **Documentation**。
  3. 在 **Description** 中输入您要提供的信息。包括文档相关部分的链接。
  4. 点 **Submit Bug**。

## 第 1 章 RED HAT ENTERPRISE LINUX 9 中的软件管理工具

在 Red Hat Enterprise Linux 9 中，DNF 工具保证软件安装。红帽继续支持使用 **yum** 术语与之前的 RHEL 主要版本保持一致。如果您键入 **yum** 而不是 **dnf**，则命令可以正常工作，因为这两者是兼容性的别名。



### 注意

虽然 RHEL 8 和 RHEL 9 基于 **DNF**，但它们与 RHEL 7 中使用的 **YUM** 兼容。

## 第 2 章 RHEL 9 发布的内容

以下小节提供了 Red Hat Enterprise Linux 9 中的软件发布概述：

- [第 2.1 节](#)。“[存储库](#)”描述如何将 Red Hat Enterprise Linux 9 的内容分成 BaseOS 和 AppStream。
- [第 2.2 节](#)。“[Application Streams](#)”描述 Application Streams 的概念。
- [第 2.3 节](#)。“[modules](#)”描述模块的概念。
- [第 2.4 节](#)。“[模块流](#)”按版本描述内容组织。
- [第 2.5 节](#)。“[模块配置集](#)”按目的描述内容组织。

### 2.1. 软件仓库

RHEL 9 内容通过两个主要软件仓库发布：**BaseOS** 和 **AppStream**。基本 RHEL 安装需要 BaseOS 和 AppStream 内容集，所有 RHEL 订阅都可以使用。有关安装说明，请参阅 [执行标准 RHEL 安装文档](#)。

#### BaseOS

BaseOS 仓库的内容旨在提供底层操作系统功能的核心组件，为所有安装提供基础操作系统的基础。这部分内容采用 RPM 格式，它的支持条款与之前的 Red Hat Enterprise Linux 版本相似。

#### AppStream

AppStream 仓库的内容包括额外的用户空间应用程序、运行时语言和数据库来支持各种工作负载和使用案例。

#### CodeReady Linux Builder

CodeReady Linux Builder 存储库在所有 RHEL 订阅中可用。它为开发人员提供了额外的软件包。不支持包括在 CodeReady Linux Builder 存储库中的软件包。

#### 其它资源

- [执行标准 RHEL 安装](#)
- [软件包清单](#)

### 2.2. 应用程序流

用户空间组件的多个版本会以 Application Streams（应用程序流）的形式提供，其更新频率会比核心操作系统软件包的更新频率更快。这为自定义 RHEL 提供了更大的灵活性，而不影响平台或特定部署的基本稳定性。

每个 Application Stream 组件都有特定的生命周期，它与 RHEL 9 或更短的生命周期相同。

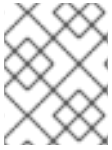
[Red Hat Enterprise Linux Application Streams 生命周期页](#) 带有较短的生命周期信息。

应用程序流以 RPM 格式提供，可以是一个模块（RPM 格式的一个扩展），软件集合（Software Collections），或 Flatpaks。

RHEL 9 提供了初始的 Application Stream 版本，可以使用传统的 **dnf install** 命令作为 RPM 软件包安装，这改进了 Application Streams 体验。

一些额外的 Application Stream 版本将作为模块发布，并在将来的次版本的 RHEL 9 版本中带有较短的生命周期。

建议查看 [Red Hat Enterprise Linux Application Stream Lifecycle](#) 定义中是否有任何内容生命周期注意事项。



### 注意

不是所有模块都是 Application Streams。其他模块的相依性不被视为 Application Streams。

### 其它资源

- [Red Hat Enterprise Linux 生命周期](#)
- [Red Hat Enterprise Linux Application Streams 生命周期](#)
- [软件包清单](#)

## 2.3. 模块

模块是一组代表组件且通常一起安装的 RPM 软件包。典型的模块包含应用的软件包、应用特定依赖项库的软件包、应用文档的软件包，以及帮助程序实用程序的软件包。



### 注意

以后的 RHEL 9 版本中将提供模块。

## 2.4. 模块流

模块流是可以作为 AppStream 物理存储库中的虚拟存储库的过滤器。模块流代表 AppStream 组件的版本。每个流都单独接收更新。

模块流可以是活跃的或者不活跃的。有效流可让系统访问特定模块流中的 RPM 软件包，以便安装相应的组件版本。如果用户操作显式启用了流，则流会活跃。

给定时间点上只能激活一个特定模块的流。因此，在一个系统中只能安装一个组件版本。不同的容器可以使用不同的版本。

每个模块都可以具有默认流。默认流可让您轻松使用 RHEL 软件包而无需了解模块。默认流是活跃的，除非禁用了整个模块或启用了该模块的另一个流。

某些模块流可以依赖于其他模块流。

要为运行时用户应用程序或开发人员应用程序选择特定的流，请考虑以下几点：

- 所需功能以及哪个组件版本支持该功能
- 兼容性
- [生命周期](#) 长度和您的更新计划

有关每个组件更改，[请查看发行注记](#)。

## 2.5. 模块配置集

---

配置集 (profile) 是为特定用例一起安装的推荐软件包列表, 比如服务器、客户端、开发、最小安装或其他用例。这些软件包列表可以包含模块流以外的软件包, 通常是来自 BaseOS 存储库或流依赖项的软件包。

使用配置集安装软件包是为方便用户提供的一次性操作。它不会阻止安装或卸载该模块提供的任何软件包。您还可以使用同一模块流的多个配置集安装软件包, 而无需进一步准备步骤。

每个模块流可以有任何数量的配置集, 包括没有。对于任何给定模块流, 其某些配置集可以被标记为 *default*, 并在没有显式指定配置集时用于配置集安装操作。但是, 对于一个模块需要有一个默认配置集不是必须的。

## 第 3 章 配置 DNF

DNF 及相关工具的配置信息保存在 `/etc/dnf/dnf.conf` 文件中。此文件包含一个必须的 `[main]` 部分，它可让您设置具有全局效果的 DNF 选项。

以下小节描述了如何进行：

- 查看当前的 DNF 配置。
- 设置 DNF `[main]` 选项。
- 使用 DNF 插件。

### 3.1. 查看当前的 DNF 配置

以下流程描述了如何显示当前 DNF 配置。

#### 流程

- 要显示在 `/etc/dnf/dnf.conf` 文件的 `[main]` 部分指定的全局 DNF 选项的当前值，请使用：

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

### 3.2. 设置 DNF 主选项

`/etc/dnf/dnf.conf` 配置文件包含一个 `[main]` 部分。本节中的键值对会影响 DNF 的运行方式并对待存储库。

您可以在 `/etc/dnf/dnf.conf` 中的 `[main]` 部分标题下添加附加选项。

有关可用 `[main]` 选项的完整列表，请查看 `dnf.conf(5)` man page 的 `[main] OPTIONS` 部分。

### 3.3. 使用 DNF 插件

DNF 提供扩展和增强操作的插件。默认安装某些插件。

以下流程描述了如何启用、配置和禁用 DNF 插件。

#### 3.3.1. 管理 DNF 插件

插件配置文件总是包含一个 `[main]` 部分，`enabled=` 选项控制在运行 `dnf` 命令时插件是否启用。如果缺少这个选项，您可以手动将其添加到该文件中。

每个安装的插件可能在 `/etc/dnf/plugins/` 目录中拥有自己的配置文件。您可以在这些文件中启用或禁用特定插件选项。

#### 3.3.2. 启用和禁用 DNF 插件

在 DNF 工具中，插件会被默认加载。

以下流程描述了如何修改 DNF 插件的加载，以及如何启用或禁用特定的 DNF 插件。

#### 流程

- 要禁用或启用 DNF 插件的加载，请确保 `/etc/dnf/dnf.conf` 文件的 `[main]` 部分有以 `plugins=` 开头的行。
  1. 要禁用 DNF 插件的加载，将 `plugins=` 的值设置为 `0`。



### 重要

不建议禁用所有插件。某些插件提供重要的 DNF 服务和命令。特别是 `product-id` 和 `subscription-manager` 插件，它们为基于证书的 **内容发布网络 (CDN)** 提供支持。全局禁用插件作为方便选项提供，建议仅在诊断 DNF 时存在潜在问题。

2. 要启用 DNF 插件的加载，请将 `plugins=` 的值设置为 `1`。
- 要禁用特定命令的所有 DNF 插件，请在该命令中附加 `--noplugins` 选项。例如，为 `update` 命令禁用 DNF 插件：

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

- 要在一个命令中禁用特定的 DNF 插件，请在命令中附加 `--disableplugin=plugin-name` 选项。例如，要为 `update` 命令禁用特定的 DNF 插件：

```
# dnf update --disableplugin=plugin-name
```

使用插件的名称替换 `plugin-name`。

- 要为单个命令启用特定的 DNF 插件，请在命令中附加 `--enableplugin=plugin-name` 选项。例如，要为 `update` 命令禁用特定的 DNF 插件：

```
# dnf update --enableplugin=plugin-name
```

使用插件的名称替换 `plugin-name`。

## 第 4 章 搜索 RHEL 9 内容

以下小节介绍了如何在 Red Hat Enterprise Linux 9 的 AppStream 和 BaseOS 软件仓库中查找和检查内容：

- 第 4.1 节：“搜索软件包” 描述了如何搜索提供所需内容的软件包。
- 第 4.2 节：“列出软件包” 描述如何列出已安装和可用的软件包。
- 第 4.3 节：“列出软件仓库” 描述如何列出启用和禁用的存储库。
- 第 4.4 节：“显示软件包信息” 描述了如何显示可用软件包的信息。
- 第 4.5 节：“列出软件包组” 描述如何列出已安装和可用的软件包组。
- 第 4.6 节：“列出可用模块” 描述了如何列出可用模块并查找有关模块的详情。
- 第 4.7 节：“指定 dnf 输入中的全局表达式” 描述了如何确保以预期将全局表达式传递给 **dnf**。

### 4.1. 搜索软件包

这部分论述了查找提供特定应用程序或其他内容的软件包所需的步骤。

#### 流程

- 要搜索软件包，使用：

```
# dnf search term
```

使用与该软件包相关的术语替换 *term*。

请注意，**dnf search** 命令返回与软件包名称和摘要中的术语匹配。这样，您可以更快地搜索搜索，您可以搜索您不知道名称但知道相关术语的软件包。

- 要在软件包描述中包含匹配名，请使用：

```
# dnf search --all term
```

用您要在软件包名称、概述或描述中搜索的术语替换 *term*。

请注意，**dnf search --all** 命令会进行更详细但速度较慢的搜索。

### 4.2. 列出软件包

以下流程描述了如何使用 **dnf** 列出可用软件包。

#### 流程

- 要列出所有安装以及可用软件包的信息，请使用：

```
# dnf list --all
```

- 要列出安装在您的系统中的所有软件包，请使用：



```
# dnf list --installed
```

或者：

```
# dnf repoquery --installed
```

- 要列出所有已启用的仓库中用于安装的所有软件包，请使用：

```
# dnf list --available
```

或者：

```
# dnf repoquery
```

请注意，您可以通过将全局表达式附加为参数来过滤结果。如需了解更多详细信息，请参阅[在 dnf 输入中指定全局表达式](#)。

### 4.3. 列出软件仓库

以下流程描述了如何使用 **dnf** 列出软件仓库。

#### 流程

- 要列出您系统中所有启用的库，请使用：

```
# dnf repolist
```

- 要列出系统中所有禁用的软件仓库，请使用：

```
# dnf repolist --disabled
```

- 要列出启用和禁用的存储库，请使用：

```
# dnf repolist --all
```

- 要列出有关存储库的附加信息，请使用：

```
# dnf repoinfo
```

请注意，您可以通过传递 ID 或库名称作为参数或者附加全局表达式来过滤结果。如需了解更多详细信息，请参阅[在 dnf 输入中指定全局表达式](#)。

### 4.4. 显示软件包信息

以下流程描述了如何使用 **dnf** 显示软件包信息。

#### 流程

- 要显示一个或多个可用软件包的信息，请使用：

```
# dnf info package-name
```

使用软件包名称替换 *package-name*。

或者：

```
# dnf repoquery --info package-name
```

使用软件包名称替换 *package-name*。

- 要显示安装在您的系统中的一个或多个软件包的信息，请使用：

```
# dnf repoquery --info --installed package-name
```

使用软件包名称替换 *package-name*。

请注意，您可以通过将全局表达式附加为参数来过滤结果。如需了解更多详细信息，请参阅[在 dnf 输入中指定全局表达式](#)。

## 4.5. 列出软件包组

以下流程描述了如何使用 **dnf** 列出软件包组。

- 要查看已安装的和可用组的数量，请使用：

```
# dnf group summary
```

- 要列出所有安装的和可用的组，请使用：

```
# dnf group list
```

请注意，您可以通过为 **dnf group list** 命令添加命令行选项 (**--hidden**、**--available**) 来过滤结果。更多可用选项请查看 man page。

- 要列出特定组群中包含的强制和可选软件包，请使用：

```
# dnf group info group-name
```

用组群的名称替换 *group-name*。

请注意，您可以通过将全局表达式附加为参数来过滤结果。如需了解更多详细信息，请参阅[在 dnf 输入中指定全局表达式](#)。

## 4.6. 列出可用模块

以下流程描述了如何查找哪些模块可用，以及它们的详细信息使用 **dnf**。

### 流程

- 列出您的系统可用的模块流：

```
# dnf module list
```

此命令的输出列出了单独行中的名称、流、配置集和概述的模块流。



## 注意

以后的 RHEL 9 版本中将提供模块。

### 其它资源

- [模块](#)
- [模块流](#)
- [模块配置集](#)

## 4.7. 在 DNF 输入中指定全局表达式

使用 `dnf` 命令，您可以将一个或多个 *全局表达式* 附加为参数来过滤结果。当作为参数传递给 `dnf` 命令时，全局表达式必须被转义。

以下规程描述了确保将全局表达式传递给 `dnf` 的两个方法，具体如预期一样。

### 流程

- 双引号或单引号包括整个全局表达式：

```
# dnf provides "*/file-name"
```

用文件名替换 *file-name*。

请注意，*文件名* 前面需要有 `/` 或 `*/` 字符序列以提供所需的结果。

- 在它们前面使用反斜杠(`\`)字符转义通配符字符：

```
# dnf provides \*/file-name
```

用文件名替换 *file-name*。

## 4.8. 其它资源

- [在 RHEL 9 中列出内容的命令](#)

## 第 5 章 安装 RHEL 9 内容

以下小节论述了如何在 Red Hat Enterprise Linux 9 中安装内容：

- [第 5.1 节"安装软件包"](#) 包括安装软件包的步骤。
- [第 5.2 节："安装软件包组"](#) 描述了如何安装软件包组。
- [第 5.3 节："运行安装内容"](#) 提供用于运行 RHEL 9 安装的内容的详情。

### 5.1. 安装软件包

以下流程描述了如何使用 **dnf** 安装软件包。

#### 流程

- 安装软件包：

```
# dnf install package-name
```

使用软件包名称替换 *package-name*。

- 要同时安装多个软件包及其依赖软件包，请使用：

```
# dnf install package-name-1 package-name-2
```

使用软件包名称替换 *package-name-1* 和 *package-name-2*。

- 当在一个 *multilib* 系统（AMD64、Intel 64 机器）中安装软件包时，您可以指定软件包的构架，方法是将其附加到软件包名称中：

```
# dnf install package-name.arch
```

使用软件包的名称和构架替换 *package-name.arch*。

- 如果您知道要安装的二进制代码的名称，但不知道软件包名，则可以使用到这个二进制代码的路径作为一个参数：

```
# dnf install /usr/sbin/binary-file
```

使用二进制文件路径替换 */usr/sbin/binary-file*。

**dnf** 搜索软件包列表，找到提供 */usr/sbin/binary-file* 的软件包，并提示您是否安装该软件包。

- 要从本地目录中安装之前下载的软件包，请使用：

```
# dnf install /path/
```

使用到该软件包的路径替换 */path/*。

### 5.2. 安装软件包组

以下流程描述了如何按组名称或带有 **dnf** 的 groupID 安装软件包组。

## 流程

- 要根据组名称安装软件包组，请使用：

```
# dnf group install group-name
```

使用组群或者环境组群的完整名称替换 *group-name*。

- 要根据 groupID 安装软件包组，请使用：

```
# dnf group install groupID
```

使用组 ID 替换 *groupID*。

## 5.3. 运行安装的内容

在从 RHEL 9 软件仓库安装内容后，通常会启用新命令。如果命令源自一个模块启用的 RPM 软件包，则使用 命令的体验应没有不同。

## 流程

- 要运行新命令，直接使用它们：

```
$ command
```

## 5.4. 其它资源

- [dnf\(8\) man page](#)
- [在 RHEL 9 中安装内容的命令](#)

## 第 6 章 更新 RHEL 9 内容

使用 **DNF**，您可以检查您的系统是否有待处理的更新。您可以列出需要更新的软件包，并选择更新单个软件包、多个软件包或者所有软件包。如果您选择更新的软件包有依赖项，它们也会被更新。

以下小节论述了如何使用 **DNF** 在 Red Hat Enterprise Linux 9 中更新内容：

- [第 6.1 节：“检查更新”](#) 描述了如何检查可用更新。
- [第 6.2 节：“更新软件包”](#) 描述如何更新单个软件包、软件包组或所有软件包及其依赖项。
- [第 6.3 节：“更新与安全性相关的软件包”](#) 描述了如何应用安全更新。

### 6.1. 检查更新

以下流程描述了如何使用 **dnf** 检查系统上安装的软件包的可用更新。

#### 流程

- 运行以下命令，以查看系统中安装的软件包是否有可用更新：

```
# dnf check-update
```

输出返回有可用更新的软件包及其依赖项列表。

### 6.2. 更新软件包

以下流程描述了如何使用 **dnf** 更新单个软件包、软件包组或所有软件包及其依赖项。

#### 流程

- 要更新所有软件包及其依赖项，请使用：

```
# dnf update
```

- 要更新单个软件包，请使用：

```
# dnf update package-name
```

使用软件包名称替换 *package-name*。

- 要更新软件包组，请使用：

```
# dnf group update group-name
```

使用软件包组的名称替换 *group-name*。



#### 重要

当对内核应用更新时，无论是否使用了 **dnf update** 或 **dnf install** 命令，**dnf** 总会安装一个新内核。

## 6.3. 更新与安全相关的软件包

以下流程描述了如何使用 **dnf** 更新与安全相关的软件包。

### 流程

- 要升级到有安全勘误的最新可用软件包，请使用：

```
# dnf update --security
```

- 要升级到最后一个安全勘误软件包，请使用：

```
# dnf update-minimal --security
```

## 第 7 章 在 RHEL 9 中自动化软件更新

要自动检查并下载软件包更新，您可以使用 **dnf-automatic** 软件包提供的 **DNF Automatic** 工具。

**DNF Automatic** 是 **DNF** 的替代命令行界面，它适用于使用 **systemd** 计时器、**cron** 作业和其他此类工具自动和常规执行。

**DNF Automatic** 根据需要同步软件包元数据，检查可用的更新，然后根据您配置工具的方式执行以下操作之一：

- **Exit**
- 下载更新的软件包
- 下载并应用更新

然后，通过所选的机制（如标准输出或电子邮件）报告操作的结果。

以下小节介绍了如何在 Red Hat Enterprise Linux 9 中自动进行软件更新：

- [第 7.1 节。"安装 DNF Automatic"](#) 描述了如何安装 **DNF Automatic** 工具。
- [第 7.2 节。"DNF Automatic 配置文件"](#) 描述 **DNF Automatic** 配置文件及其包括的部分。
- [第 7.3 节。"启用 DNF Automatic"](#) 描述了如何启用 **DNF Automatic** 工具。
- [第 7.4 节。"包括在 \*\*dnf-automatic\*\* 软件包的 \*\*systemd timer\*\* 单元的概述"](#) 列出了 **dnf-automatic** **systemd timer** 单元

### 7.1. 安装 DNF AUTOMATIC

以下流程描述了如何安装 **DNF Automatic** 工具。

#### 流程

- 安装 **dnf-automatic** 软件包：

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

#### 验证

- 通过确认 **dnf-automatic** 软件包是否存在，来验证安装是否成功：

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

### 7.2. DNF AUTOMATIC 配置文件

默认情况下，**DNF Automatic** 使用 **/etc/dnf/automatic.conf** 作为其配置文件来定义其行为。

配置文件被分隔为以下主题部分：

- **[commands]** 部分  
设置 **DNF Automatic** 的操作模式。



- **[emitters]** 部分  
定义如何报告 DNF Automatic 的结果。
- **[command\_email]** 部分  
为用来发送电子邮件的外部命令提供电子邮件发布程序配置。
- **[email]** 部分  
提供电子邮件发布程序配置。
- **[base]** 部分  
覆盖 DNF 的主要配置文件中的设置。

通过 `/etc/dnf/automatic.conf` 文件的默认设置，DNF Automatic 检查可用更新，下载它们，并将结果报告为标准输出。



### 警告

**[commands]** 部分中的操作模式设置会被 `systemd` 定时器单元用于除 `dnf-automatic.timer` 之外的所有定时器单元使用的设置覆盖。

### 其它资源

- [DNF Automatic 文档](#)
- `man dnf-automatic` man page
- [dnf-automatic 软件包中包含的 systemd 定时器单元概述](#)

## 7.3. 启用 DNF AUTOMATIC

要运行 DNF Automatic，必须始终启用并启动特定的 `systemd` 定时器单元。您可以使用 `dnf-automatic` 软件包提供的一个定时器单元，也可以根据自己的需要写入您自己的定时器单元。

以下流程描述了如何启用 DNF Automatic。

### 先决条件

- 您可以通过修改 `/etc/dnf/automatic.conf` 配置文件来指定 DNF Automatic 的行为。

### 流程

- 要选择、启用并启动 下载 可用更新的 `systemd` 定时器单元，请使用：

```
# systemctl enable dnf-automatic-download.timer
# systemctl start dnf-automatic-download.timer
```

- 要选择、启用并启动一个用来 下载并安装 可用更新的 `systemd` 定时器单元，请使用：

```
# systemctl enable dnf-automatic-install.timer
```

-

```
# systemctl start dnf-automatic-install.timer
```

- 要选择、启用并启动 **报告** 可用更新的 systemd 计时器单元，请使用：

```
# systemctl enable dnf-automatic-notifyonly.timer
```

```
# systemctl start dnf-automatic-notifyonly.timer
```

- 要选择、启用并启动一个 systemd 计时器单元，该单元 **下载**、**下载并安装**，或者 **报告** 可用更新，请使用：

```
# systemctl enable dnf-automatic.timer
```

```
# systemctl start dnf-automatic.timer
```

- 另外，使用 **--now** 选项，选择、启用和启动 systemd 定时器单元。例如：

```
# systemctl enable --now dnf-automatic-download.timer
```



### 注意

您还可以从命令行或自定义脚本直接执行 `/usr/bin/dnf-automatic` 文件来运行 DNF Automatic。

### 验证

- 验证计时器是否已启用：

```
# systemctl status <systemd timer unit>
```

### 其它资源

- [man dnf-automatic man page](#)
- [dnf-automatic 软件包中包含的 systemd 计时器单元概述](#)
- [DNF Automatic 配置文件](#)

## 7.4. DNF-AUTOMATIC 软件包中包含的 SYSTEMD 计时器单元概述

systemd 定时器单元具有优先权，并在下载和应用更新时覆盖 `/etc/dnf/automatic.conf` 配置文件中的设置。

例如，如果您设置了：

```
download_updates = yes
```

在 `/etc/dnf/automatic.conf` 配置文件中，您已激活 `dnf-automatic-notifyonly.timer` 单元，软件包不会被下载。

**dnf-automatic** 软件包包括以下 systemd 计时器单元：

表 7.1. **dnf-automatic** 软件包中包含的 systemd 计时器

计时器单元	功能	覆盖 <code>/etc/dnf/automatic.conf</code> 文件中的设置？
<b>dnf-automatic-download.timer</b>	<p>下载软件包以便进行更新。</p> <p>注：这个计时器单元没有安装更新的软件包。要执行安装，您必须执行 <b>dnf update</b> 命令。</p>	是
<b>dnf-automatic-install.timer</b>	<p>下载并安装更新的软件包。</p>	是
<b>dnf-automatic-notifyonly.timer</b>	<p>只下载存储库数据，以保持存储库缓存最新状态，并通知您是否有可用的更新。</p> <p>注：这个计时器单元不下载或安装更新的软件包</p>	是
<b>dnf-automatic.timer</b>	<p>此计时器在下载和应用更新时的行为是由 <code>/etc/dnf/automatic.conf</code> 配置文件中的设置指定。</p> <p>默认行为与 <b>dnf-automatic-download.timer</b> 单元相同：它会下载软件包，但不安装它们。</p>	否

#### 其它资源

- **man dnf-automatic manual** man page
- [DNF Automatic 配置文件](#)

## 第 8 章 删除 RHEL 9 内容

以下小节介绍了如何在 Red Hat Enterprise Linux 9 中删除内容：

- [第 8.1 节。](#)“[删除安装的软件包](#)” 描述删除软件包。
- [第 8.2 节。](#)“[删除软件包组](#)” 描述删除软件包组。

### 8.1. 删除安装的软件包

以下流程描述了如何使用 `dnf` 删除软件包。

#### 流程

- 要删除某个软件包以及所有未使用的依赖软件包，请使用：

```
# dnf remove package-name
```

使用软件包名称替换 `package-name`。

请注意，软件包会与其他依赖软件包一起删除。

- 要同时删除多个软件包及其未使用的依赖项，请使用：

```
# dnf remove package-name-1 package-name-2
```

使用软件包名称替换 `package-name-1` 和 `package-name-2`。



#### 注意

`dnf` 在删除依赖软件包的情况下无法删除软件包。

### 8.2. 删除软件包组

以下流程描述了如何根据组群名称或 `groupID` 删除软件包。

#### 流程

- 要根据组群名称删除软件包组，请使用：

```
# dnf group remove group-name
```

使用组群的全名替换 `group-name`。

- 要通过 `groupID` 删除软件包组，请使用：

```
# dnf group remove groupID
```

使用组 ID 替换 `groupID`。

### 8.3. 其它资源

- 在 RHEL 9 中删除内容的命令

## 第 9 章 处理软件包管理历史记录

使用 **dnf history** 命令，您可以查看以下信息

- DNF 事务的时间线
- 事务发生的日期和时间
- 受事务影响的软件包数量
- 事务是成功还是被中止
- 如果在事务间更改了 RPM 数据库

**dnf history** 命令也可用于撤销或重新执行事务。

下面的部分论述了如何使用 **dnf** 在 Red Hat Enterprise Linux 9 中处理软件包管理历史记录：

- [第 9.1 节。"列出事务"](#) 描述如何列出最新事务、所选软件包的最新操作以及特定事务的详情。
- [第 9.2 节。"恢复事务"](#) 描述如何恢复选定的事务或最后一个事务。
- [第 9.3 节。"重复事务"](#) 描述如何重复选定或最后一个事务。

### 9.1. 列出事务

以下流程描述了如何列出最新的 DNF 事务、所选软件包的最新操作和特定事务的详细信息。

#### 流程

- 要显示所有最新的 DNF 事务列表，请使用：

```
# dnf history
```

- 要显示所选软件包的最新操作列表，请使用：

```
# dnf history list package-name
```

使用软件包名称替换 *package-name*。您可以通过附加全局表达式来过滤命令输出。如需了解更多信息，请参阅[在 dnf 输入中指定全局表达式](#)。

- 要显示特定事务的详情，请使用：

```
# dnf history info transactionID
```

用事务的 ID 替换 *transactionID*。

### 9.2. 恢复事务

以下流程描述了如何使用 **dnf** 恢复所选事务或最后一个事务。

#### 流程

- 要恢复特定的事务，请使用：

```
# dnf history undo transactionID
```

用事务的 ID 替换 *transactionID*。

- 要恢复到最后的事务，请使用：

```
# dnf history undo last
```

请注意，**dnf history undo** 命令仅恢复事务期间执行的步骤。如果事务安装了新的软件包，**dnf history undo** 会卸载它。如果事务卸载了软件包，**dnf historydo** 会重新安装它。如果旧软件包仍然可用，**dnf history undo** 命令还会尝试将所有更新的软件包降级到之前的版本。

### 9.3. 重复事务

以下流程描述了如何使用 **dnf** 重复选定的事务或最后一个事务。

#### 流程

- 要重复特定的事务，请使用：

```
# dnf history redo transactionID
```

用事务的 ID 替换 *transactionID*。

- 要重复最后的事务，请使用：

```
# dnf history redo last
```

请注意，**dnf history redo** 命令仅重复事务期间执行的步骤。

## 第 10 章 管理自定义软件存储库

DNF 及相关工具的配置信息保存在 `/etc/dnf/dnf.conf` 文件中。此文件包含一个或多个 `[repository]` 部分，可用于设置特定于存储库的选项。

建议您在 `/etc/yum.repos.d/` 目录中的新的或现有 `.repo` 文件中定义单独的存储库。

请注意，您在 `/etc/dnf/dnf.conf` 文件的单独 `[repository]` 部分定义的值会覆盖 `[main]` 部分中设置的值。

以下小节论述了如何在 Red Hat Enterprise Linux 9 中管理自定义软件存储库：

- 第 10.1 节。“设置 DNF 存储库选项”描述了如何设置 `[repository]` 选项。
- 第 10.2 节。“添加 DNF 存储库”描述了如何定义新的 DNF 存储库。
- 第 10.3 节。“启用 DNF 存储库”描述了如何在系统中启用 DNF 存储库。
- 第 10.4 节。“禁用 DNF 存储库”描述了如何禁用添加到您的系统中的 DNF 存储库。

### 10.1. 设置 DNF 软件仓库选项

`/etc/dnf/dnf.conf` 配置文件包含 `[repository]` 部分，其中 `repository` 是唯一的存储库 ID。`[repository]` 部分允许您定义独立的 DNF 软件仓库。



#### 注意

不要给出红帽软件仓库使用的自定义软件仓库名称以避免冲突。

有关可用 `[repository]` 选项的完整列表，请查看 `dnf.conf(5)` man page 的 `[repository] OPTIONS` 部分。

### 10.2. 添加 DNF 软件仓库

要定义新存储库，您可以：

- 在 `/etc/dnf/dnf.conf` 文件中添加 `[repository]` 部分。
- 在 `/etc/yum.repos.d/` 目录中的 `.repo` 文件中添加 `[repository]` 部分。安装 RPM 或软件管理工具（例如，订阅管理器）可以提供自己的 `.repo` 文件。



#### 注意

在 `.repo` 文件中而不是 `/etc/dnf/dnf.conf` 中定义您的仓库，作为在这个目录中带有 `.repo` 文件扩展的所有文件，则 `dnf` 会读取。

以下流程描述了如何在系统中添加 DNF 软件仓库。

#### 流程

- 在您的系统中添加软件仓库：

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



使用指向库的 URL 替换 `repository_url`。



### 警告

从基于证书的 **Content Delivery Network (CDN)** 以外的未验证或不受信任的源获得和安装软件包会带来潜在的安全风险，并可能导致安全性、稳定性、兼容性和维护问题。

## 10.3. 启用 DNF 软件仓库

以下流程描述了如何启用添加到您的系统中的 DNF 软件仓库。

### 流程

- 启用存储库：

```
# dnf-config-manager --enable repositoryID
```

使用唯一的存储库 ID 替换 `repositoryID`。

有关如何列出可用存储库 ID 的更多信息，请参阅 [列出软件包](#)。

## 10.4. 禁用 DNF 软件仓库

以下流程描述了如何禁用添加到您的系统中的 DNF 软件仓库。

### 流程

- 禁用软件仓库：

```
# dnf-config-manager --disable repositoryID
```

使用唯一的存储库 ID 替换 `repositoryID`。

有关如何列出可用存储库 ID 的更多信息，请参阅 [列出软件包](#)。

## 附录 A. DNF 命令列表

本章列出了 DNF 命令在 Red Hat Enterprise Linux 9 中列出、安装和删除内容。

### A.1. 在 RHEL 9 中列出内容的命令

下表列出了在 RHEL 9 中查找内容及其详情的常用 DNF 命令：

命令	描述
<b>dnf search <i>term</i></b>	使用与软件包相关的术语搜索软件包
<b>dnf repoquery <i>package</i></b>	为所选软件包搜索可用的 DNF 软件仓库
<b>dnf list</b>	列出所有已安装和可用软件包的信息
<b>dnf list --installed</b> <b>dnf repoquery --installed</b>	列出系统上安装的所有软件包
<b>dnf list --available</b> <b>dnf repoquery</b>	列出所有已启用的仓库中用于安装的所有软件包
<b>dnf repolist</b>	列出系统中所有启用的软件仓库
<b>dnf repolist --disabled</b>	列出系统中所有禁用的软件仓库
<b>dnf repolist --all</b>	列出启用和禁用的存储库
<b>dnf repoinfo</b>	列出仓库的附加信息
<b>dnf info <i>package-name</i></b> <b>dnf repoquery --info <i>package_name</i></b>	显示可用软件包的详情
<b>dnf repoquery --info --installed <i>package_name</i></b>	显示系统上安装的软件包详情
<b>dnf module list</b>	列出模块及其当前状态  请注意，如果软件包在任何模块外可用，这个命令的输出为空。
<b>dnf group summary</b>	查看已安装和可用的组的数量
<b>dnf group list</b>	列出所有已安装和可用的组
<b>dnf group info <i>group-name</i></b>	列出特定组中包含的强制和可选软件包

## A.2. 在 RHEL 9 中安装内容的命令

下表列出了在 RHEL 9 中安装内容的常用 DNF 命令：

命令	描述
<b>dnf install <i>package-name</i></b>	<p>安装软件包。</p> <p>如果软件包由模块流提供，<b>dnf</b> 会解析所需的模块流，并在安装此软件包时自动启用它。这也会以递归方式对所有软件包依赖项进行。如果更多模块流满足要求，则使用默认模块流。</p>
<b>dnf install <i>package-name-1 package-name-2</i></b>	同时安装多个软件包及其依赖项
<b>dnf install <i>package-name.arch</i></b>	在多 <i>lib</i> 系统中安装软件包（AMD64、Intel 64 机器）时，指定软件包的架构。
<b>dnf install <i>/usr/sbin/binary-file</i></b>	使用到二进制文件的路径作为参数安装二进制文件
<b>dnf install <i>/path/</i></b>	从本地目录安装之前下载的软件包
<b>dnf install <i>package-url</i></b>	<p>使用软件包 URL 安装远程软件包</p> <p>当您想要让软件包可供系统使用，但目前不想安装其中任何软件包时，启用该模块。</p> <p>有些模块可能无法定义默认流。在这种情况下，您必须明确指定流。</p> <p>如果该模块定义了默认流，您可以省略 <i>stream</i> 和冒号。</p> <p>请注意，运行这个命令不会安装任何 RPM 软件包。</p> <p>请注意，有些模块不定义默认流。</p> <p>请注意，运行这个命令还会启用指定的流。</p>
<b>dnf group install <i>group-name</i></b>	按组名称安装软件包组
<b>dnf group install <i>groupID</i></b>	根据 <i>groupID</i> 安装软件包组

## A.3. 在 RHEL 9 中删除内容的命令

下表列出了在 RHEL 9 中删除内容的常用 DNF 命令：

命令	描述
<b>dnf remove <i>package-name</i></b>	删除特定软件包以及所有依赖软件包

命令	描述
<b>dnf remove <i>package-name-1 package-name-2</i></b>	同时删除多个软件包及其未使用的依赖项
<b>dnf group remove <i>group-name</i></b>	按组名称删除软件包组
<b>dnf group remove <i>groupID</i></b>	<p>按 groupID 删除软件包组</p> <p>请注意，运行此命令可从系统中删除关键软件包。</p> <p>请注意，运行此命令不会从指定的模块中删除软件包。</p> <p>请注意，运行此命令不会从指定的模块中删除软件包。</p>