



Red Hat Enterprise Linux 9

DNF ツールを使用したソフトウェアの管理

Red Hat Enterprise Linux 9 の DNF によるソフトウェア管理のガイド

Red Hat Enterprise Linux 9 DNF ツールを使用したソフトウェアの管理

Red Hat Enterprise Linux 9 の DNF によるソフトウェア管理のガイド

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.

法律上の通知

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[®] 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 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.

概要

本ドキュメントでは、Red Hat Enterprise Linux 9 の DNF ツールを使用した AppStream および BaseOS リポジトリ内のコンテンツの検索、発見、インストール、および使用について説明します。これには、モジュール、アプリケーションストリーム、およびプロファイルの使用方法が含まれます。

目次

| | |
|---|----|
| 多様性を受け入れるオープンソースの強化 | 4 |
| RED HAT ドキュメントへのフィードバック (英語のみ) | 5 |
| 第1章 RED HAT ENTERPRISE LINUX 9 のソフトウェア管理ツール | 6 |
| 第2章 RHEL 9 のコンテンツの配布 | 7 |
| 2.1. リポジトリ | 7 |
| 2.2. アプリケーションストリーム | 7 |
| 2.3. MODULES | 8 |
| 2.4. モジュールストリーム | 8 |
| 2.5. モジュールプロファイル | 9 |
| 第3章 DNF の設定 | 10 |
| 3.1. 現在の DNF 設定の表示 | 10 |
| 3.2. DNF メインオプションの設定 | 10 |
| 3.3. DNF プラグインの使用 | 10 |
| 3.3.1. DNF プラグインの管理 | 10 |
| 3.3.2. DNF プラグインの有効化および無効化 | 10 |
| 第4章 RHEL 9 コンテンツの検索 | 12 |
| 4.1. ソフトウェアパッケージの検索 | 12 |
| 4.2. ソフトウェアパッケージの一覧表示 | 12 |
| 4.3. リポジトリの一覧表示 | 13 |
| 4.4. パッケージ情報の表示 | 14 |
| 4.5. パッケージグループの一覧表示 | 14 |
| 4.6. 利用可能なモジュールの一覧表示 | 15 |
| 4.7. DNF INPUT での GLOB 表現の指定 | 15 |
| 4.8. 関連情報 | 15 |
| 第5章 RHEL 9 コンテンツのインストール | 17 |
| 5.1. パッケージのインストール | 17 |
| 5.2. パッケージグループのインストール | 18 |
| 5.3. インストール済みコンテンツの実行 | 18 |
| 5.4. 関連情報 | 18 |
| 第6章 RHEL 9 コンテンツの更新 | 19 |
| 6.1. 更新の確認 | 19 |
| 6.2. パッケージの更新 | 19 |
| 6.3. セキュリティー関連パッケージの更新 | 20 |
| 第7章 RHEL 9 でのソフトウェア更新の自動化 | 21 |
| 7.1. DNF AUTOMATIC のインストール | 21 |
| 7.2. DNF AUTOMATIC 設定ファイル | 21 |
| 7.3. DNF AUTOMATIC の有効化 | 22 |
| 7.4. DNF-AUTOMATIC に含まれる SYSTEMD タイマーユニットの概要 | 23 |
| 第8章 RHEL 9 コンテンツの削除 | 26 |
| 8.1. インストール済みパッケージの削除 | 26 |
| 8.2. パッケージグループの削除 | 26 |
| 8.3. 関連情報 | 27 |
| 第9章 パッケージ管理履歴の処理 | 28 |
| 9.1. トランザクションの一覧表示 | 28 |

| | |
|--------------------------------------|-----------|
| 9.2. トランザクションの復元 | 29 |
| 9.3. トランザクションの繰り返し | 29 |
| 第10章 カスタムソフトウェアリポジトリの管理 | 30 |
| 10.1. DNF リポジトリオプションの設定 | 30 |
| 10.2. DNF リポジトリの追加 | 30 |
| 10.3. DNF リポジトリの有効化 | 31 |
| 10.4. DNF リポジトリの無効化 | 31 |
| 付録A DNF コマンド一覧 | 32 |
| A.1. RHEL 9 でコンテンツを一覧表示するコマンド | 32 |
| A.2. RHEL 9 にコンテンツをインストールするコマンド | 33 |
| A.3. RHEL 9 でコンテンツを削除するコマンド | 34 |

多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、[Red Hat CTO である Chris Wright のメッセージ](#)をご覧ください。

RED HAT ドキュメントへのフィードバック (英語のみ)

ご意見ご要望をお聞かせください。ドキュメントの改善点はございませんか。

- 特定の部分についての簡単なコメントをお寄せいただく場合は、以下をご確認ください。
 1. ドキュメントの表示が **Multi-page HTML** 形式になっていて、ドキュメントの右上隅に **Feedback** ボタンがあることを確認してください。
 2. マウスカーソルで、コメントを追加する部分を強調表示します。
 3. そのテキストの下に表示される **Add Feedback** ポップアップをクリックします。
 4. 表示される手順に従ってください。
- Bugzilla を介してフィードバックを送信するには、新しいチケットを作成します。
 1. [Bugzilla](#) の Web サイトに移動します。
 2. Component で **Documentation** を選択します。
 3. **Description** フィールドに、ドキュメントの改善に関するご意見を記入してください。ドキュメントの該当部分へのリンクも記入してください。
 4. **Submit Bug** をクリックします。

第1章 RED HAT ENTERPRISE LINUX 9 のソフトウェア管理ツール

Red Hat Enterprise Linux 9 では、ソフトウェアのインストールを確実に行うための **DNF** ツールを使用します。Red Hat は、RHEL の以前のメジャーバージョンとの一貫性のために、**yum** という用語の使用を引き続きサポートします。**dnf** の代わりに **yum** と入力しても、どちらも互換性のためのエイリアスなので、コマンドは想定どおりに動作します。



注記

RHEL 8とRHEL 9は**DNF**をベースにしていますが、RHEL 7で使用していた**YUM**との互換性があります。

第2章 RHEL 9 のコンテンツの配布

以下のセクションでは、Red Hat Enterprise Linux 9 におけるソフトウェアディストリビューションの概要を説明します。

- [セクション 2.1.リポジトリ](#)では、Red Hat Enterprise Linux 9 のコンテンツが BaseOS および AppStream にどのように分割されるかを説明します。
- [セクション 2.2.アプリケーションストリーム](#)では、アプリケーションストリームの概念を説明します。
- [セクション 2.3.“Modules”](#) では、モジュールの概念を説明します。
- [セクション 2.4.「モジュールストリーム」](#) では、コンテンツのバージョン別の編成を説明します。
- [セクション 2.5.「モジュールプロファイル」](#) では、コンテンツの目的別の編成を説明します。

2.1. リポジトリ

RHEL 9 のコンテンツは、2つのメインリポジトリにより配布されます。**BaseOS** および **AppStream**。基本的な RHEL インストールには、BaseOS と AppStream の両方のコンテンツセットが必要で、すべての RHEL サブスクリプションで利用できます。インストール手順は『[標準的な RHEL 9 インストールの実行](#)』を参照してください。

BaseOS

BaseOS リポジトリのコンテンツは、すべてのインストールの基盤となる、基本的な OS 機能のコアセットを提供します。このコンテンツは RPM 形式で提供されており、以前のリリースの Red Hat Enterprise Linux と同様のサポート条件が適用されます。

AppStream

AppStream リポジトリには、さまざまなワークロードとユースケースに対応するために、ユーザー空間アプリケーション、ランタイム言語、およびデータベースが同梱されます。

CodeReady Linux Builder

CodeReady Linux Builder リポジトリは、すべての RHEL サブスクリプションで利用できます。このリポジトリは、開発者向けの追加パッケージを提供します。CodeReady Linux Builder リポジトリに含まれるパッケージには対応しません。

関連情報

- [標準的な RHEL 9 インストールの実行](#)
- [パッケージマニフェスト](#)

2.2. アプリケーションストリーム

複数のバージョンのユーザー空間コンポーネントが Application Streams として提供され、コアオペレーティングシステムパッケージよりも頻繁に更新されます。これにより、プラットフォームや特定のデプロイメントの基盤となる安定性に影響を及ぼさずに、RHEL をより柔軟にカスタマイズできます。

アプリケーションストリームの各コンポーネントには、RHEL9と同じか、特定のアプリケーションに適した短いライフサイクルが設定されています。RHEL のライフサイクルの詳細は、「[Red Hat Enterprise Linux のライフサイクル](#)」および「[Red Hat Enterprise Linux Application Streams ライフサイクル](#)」を参照してください。

Application Streams は以下の形式で利用できます。

- 使い慣れた RPM 形式
- モジュールと呼ばれる RPM 形式の拡張
- Software Collections
- Flatpaks として実行します。

RHEL 9 では、従来の **dnf install** コマンドを使用して、RPM パッケージとしてインストールできる初めての Application Stream バージョンを提供することで、Application Streams エクスペリエンスを向上させています。



注記

RPM 形式の一部の初期 Application Streams のライフサイクルは、Red Hat Enterprise Linux 9 よりも短くなります。

アプリケーションストリームの一部の追加バージョンは、今後の RHEL 9 のマイナーリリースで、ライフサイクルの短いモジュールとして配布される予定です。

インストールする Application Stream のバージョンを常に判別し、最初に [Red Hat Enterprise Linux Application Stream Lifecycle](#) を確認してください。



注記

すべてのモジュールがアプリケーションストリームというわけではありません。その他のモジュールの依存関係は、アプリケーションストリームコンポーネントとは見なされません。

関連情報

- [Red Hat Enterprise Linux Life Cycle](#)
- [Red Hat Enterprise Linux Application Streams Life Cycle](#)
- [Red Hat Enterprise Linux 9:アプリケーションの互換性ガイド](#)
- [パッケージマニフェスト](#)

2.3. MODULES

モジュールはコンポーネントを表し、通常はまとめてインストールされる一連の RPM パッケージです。通常のモジュールには、アプリケーションを含むパッケージ、アプリケーション固有の依存関係ライブラリーを含むパッケージ、アプリケーションのドキュメントを含むパッケージ、およびヘルプユーティリティを含むパッケージが含まれます。



注記

モジュールは、今後の RHEL 9 のマイナーリリースで利用できます。

2.4. モジュールストリーム

モジュールのストリームは、AppStream 物理リポジトリで仮想的なリポジトリとして扱えるフィルターです。モジュールストリームは、AppStream コンポーネントのバージョンを表します。各ストリームが、それぞれ独立して更新を受け取ります。

モジュールストリームは、アクティブまたは非アクティブにできます。アクティブなストリームでは、特定モジュールストリームの RPM パッケージにシステムがアクセスできるようになり、コンポーネントの各バージョンのインストールを可能にします。ストリームは、ユーザーのアクションによって明示的に有効化された場合にアクティブになります。

あるモジュールの中で同時にアクティブにできるストリームは1つだけです。したがって、1つのシステムにインストールできるコンポーネントは、1つのバージョンだけです。複数のコンテナで異なるバージョンを使用できます。

各モジュールは、デフォルトのストリームを持つことができます。デフォルトのストリームは、モジュールについて学ぶ必要はなく、通常の方法で RHEL パッケージを簡単に使用できます。モジュール全体が無効になっている場合、またはそのモジュールの別のストリームが有効になっている場合を除き、デフォルトのストリームはアクティブです。

特定のモジュールストリームは、他のモジュールストリームに依存できます。

ランタイムユーザーアプリケーションまたは開発者用アプリケーションの特定のストリームを選択するには、以下を検討してください。

- 必要な機能と、その機能に対応するコンポーネントのバージョン
- 互換性
- [ライフサイクル](#) の期間と更新計画

コンポーネントごとの変更は、[リリースノート](#) を参照してください。

2.5. モジュールプロファイル

プロファイルとは、サーバー、クライアント、開発、最小限のインストールなど、特定のユースケースと一緒にインストールすることを推奨するパッケージのリストです。このパッケージ一覧は、モジュールストリームに含まれないパッケージ (通常は BaseOS リポジトリ、またはそのストリームの依存関係) も含みます。

利便性のため、プロファイル (一度の操作で完了) を利用してパッケージのインストールを行えます。そのモジュールが提供する任意のパッケージをインストールおよびアンインストールすることへの影響はありません。同じモジュールストリームから複数のプロファイルを利用してパッケージをインストールすることも、特に準備などを行わずに行えます。

各モジュールストリームではプロファイル数に制限がありません (ゼロにすることも可能)。モジュールストリームでは、プロファイルの一部が **デフォルト** となり、その他のプロファイルが明示的に指定されていない場合にインストールに使用されるプロファイルとなります。ただし、モジュールストリームのデフォルトプロファイルは必須ではありません。

第3章 DNF の設定

DNF および関連ユーティリティーの設定情報は、`/etc/dnf/dnf.conf` ファイルに保存されます。このファイルには、グローバルに影響する DNF オプションを設定できる必須の `[main]` セクションが1つ含まれています。

以下のセクションでは、次の方法を説明します。

- 現在の DNF 設定を表示します。
- DNF `[main]` オプションを設定します。
- DNF プラグインを使用します。

3.1. 現在の DNF 設定の表示

以下の手順では、現在の DNF 設定の表示方法を説明します。

手順

- `/etc/dnf/dnf.conf` ファイルの `[main]` セクションで指定されるグローバル DNF オプションの現在の値を表示するには、以下を使用します。

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

3.2. DNF メインオプションの設定

`/etc/dnf/dnf.conf` 設定ファイルには `[main]` セクションが1つ含まれています。本セクションのキーと値のペアは、DNF のリポジトリの操作方法および処理方法に影響を及ぼします。

`/etc/dnf/dnf.conf` の `[main]` セクションの下に、オプションを追加できます。

利用可能な `[main]` オプションの完全なリストは、`dnf.conf(5)` man ページの `[main] OPTIONS` セクションを参照してください。

3.3. DNF プラグインの使用

DNF は、その操作を拡張および強化するプラグインを提供します。特定のプラグインが、デフォルトでインストールされています。

以下の手順では、DNF プラグインを有効、設定、および無効にする方法を説明します。

3.3.1. DNF プラグインの管理

プラグインの設定ファイルには必ず `[main]` セクションがあり、`enabled=` オプションは、`dnf` コマンドを実行する際にプラグインを有効にするかどうかを制御します。このオプションがファイルに含まれていない場合は手動で追加できます。

インストールした各プラグインの `/etc/dnf/plugins/` ディレクトリーには、それぞれ独自の設定ファイルが含まれている場合があります。これらのファイルでは、プラグイン固有のオプションを有効または無効にできます。

3.3.2. DNF プラグインの有効化および無効化

DNF ツールでは、プラグインはデフォルトで読み込まれます。

以下の手順では、DNF プラグインの読み込みを変更し、特定の DNF プラグインを有効または無効にする方法を説明します。

手順

- DNF プラグインの読み込みを無効または有効にするには、`/etc/dnf/dnf.conf` ファイルの `[main]` セクションに、`plugins=` で始まる行が存在することを確認します。

1. DNF プラグインの読み込みを無効にするには、`plugins=` の値を `0` に設定します。



重要

プラグインをすべて無効にすることは推奨していません。プラグインには、重要な DNF サービスおよびコマンドを提供するものがあります。その中でも `product-id` プラグインおよび `subscription-manager` プラグインは、証明書ベースの **Content Delivery Network (CDN)** への対応に必要です。グローバルにプラグインを無効にすることは、便利なオプションとして提供されていますが、DNF の潜在的な問題を診断する場合にのみ推奨されます。

2. DNF プラグインの読み込みを有効にするには、`plugins=` の値を `1` に設定します。

- 特定のコマンドですべての DNF プラグインを無効にするには、コマンドに `--noplugins` オプションを追加します。たとえば、`update` コマンドで DNF プラグインを無効にするには、以下のコマンドを実行します。

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

- 1つのコマンドで特定の DNF プラグインを無効にするには、コマンドに `--disableplugin=plugin-name` オプションを追加します。たとえば、`update` コマンドで特定の DNF プラグインを無効にするには、以下のコマンドを実行します。

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

`plugin-name` をプラグインの名前に置き換えます。

- 1つのコマンドで特定の DNF プラグインを有効にするには、コマンドに `--enableplugin=plugin-name` オプションを追加します。たとえば、`update` コマンドで特定の DNF プラグインを無効にするには、以下のコマンドを実行します。

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

`plugin-name` をプラグインの名前に置き換えます。

第4章 RHEL 9 コンテンツの検索

次のセクションでは、Red Hat Enterprise Linux 9 の AppStream リポジトリおよび BaseOS リポジトリでコンテンツの場所を特定して調べる方法を説明します。

- [セクション 4.1 「ソフトウェアパッケージの検索」](#) では、必要なコンテンツを提供するパッケージを検索する方法を説明します。
- [セクション 4.2 「ソフトウェアパッケージの一覧表示」](#) では、インストールされているパッケージと、利用可能なパッケージを一覧表示する方法を説明します。
- [セクション 4.3 「リポジトリの一覧表示」](#) では、有効および無効のリポジトリを一覧表示する方法を説明します。
- [セクション 4.4 「パッケージ情報の表示」](#) では、利用可能なパッケージの情報を表示する方法を説明します。
- [セクション 4.5 「パッケージグループの一覧表示」](#) では、インストールされているパッケージグループと、利用可能なパッケージグループを一覧表示する方法を説明します。
- [セクション 4.6 「利用可能なモジュールの一覧表示」](#) では、利用可能なモジュールの一覧表示方法と、その詳細を確認する方法を説明します。
- [セクション 4.7 「dnf input での glob 表現の指定」](#) では、意図したとおりに glob 表現を確実に **dnf** に渡す方法を説明します。

4.1. ソフトウェアパッケージの検索

ここでは、特定のアプリケーションやその他のコンテンツを提供するパッケージの検索に必要な手順を説明します。

手順

- パッケージを検索するには、以下を使用します。

```
# dnf search term
```

term は、パッケージ関連の用語に置き換えます。

dnf search コマンドは、パッケージの名前と概要に含まれる用語の一致を返すことに注意してください。これにより検索時間が短縮され、名前が分からないものの、関連用語が分かっているパッケージを検索できます。

- パッケージの説明に一致する用語を含めるには、以下を使用します。

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

term は、パッケージ名、概要、または説明で検索する用語に置き換えます。

dnf search --all コマンドを使用すると、より詳細な検索が可能になりますが、検索にかかる時間が長くなることに注意してください。

4.2. ソフトウェアパッケージの一覧表示

以下の手順では、**dnf** で利用可能なパッケージを一覧表示する方法を説明します。

手順

- インストール済みで利用可能なパッケージの情報をすべて表示するには、次のコマンドを実行します。

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

- システムにインストールされているパッケージの一覧を表示するには、以下のコマンドを使用します。

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

または、次のようになります。

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

- 有効なすべてのリポジトリで、インストール可能な全パッケージを表示するには、以下を使用します。

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

または、次のようになります。

```
# dnf repoquery
```

glob 表現を引数として追加して結果をフィルターできることに注意してください。詳細は、[dnf input での glob 表現の指定](#) を参照してください。

4.3. リポジトリの一覧表示

以下の手順では、**dnf** でリポジトリを一覧表示する方法を説明します。

手順

- システムで有効なリポジトリをすべて一覧表示するには、以下を使用します。

```
# dnf repolist
```

- システムで無効になっているリポジトリをすべて一覧を表示するには、以下を使用します。

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

- 有効および無効なリポジトリの両方を一覧表示するには、以下を使用します。

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

- リポジトリに関する追加情報を一覧表示するには、以下を使用します。

```
# dnf repoinfo
```

リポジトリの ID または名前を引数として渡すか、glob 表現を追加して結果をフィルタリングできにすることに注意してください。詳細は、[dnf input での glob 表現の指定](#) を参照してください。

4.4. パッケージ情報の表示

ここでは、**dnf** を使用して、パッケージ情報を表示する方法を説明します。

手順

- 利用可能なパッケージの情報を表示するには、以下のコマンドを実行します。

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

`package-name` を、パッケージ名に置き換えます。

または、次のようになります。

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

`package-name` を、パッケージ名に置き換えます。

- システムにインストールされている1つまたは複数のパッケージの情報を表示するには、以下のコマンドを実行します。

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

`package-name` を、パッケージ名に置き換えます。

glob 表現を引数として追加して結果をフィルターできることに注意してください。詳細は、[dnf input](#) での [glob 表現の指定](#) を参照してください。

4.5. パッケージグループの一覧表示

以下の手順では、**dnf** を使用してパッケージグループの一覧を表示する方法を説明します。

- インストール済みおよび利用可能なグループの数を表示するには、以下を使用します。

```
# dnf group summary
```

- インストール済みおよび利用可能なグループをすべて一覧表示するには、以下のコマンドを使用します。

```
# dnf group list
```

dnf group list コマンドにコマンドラインオプション (**--hidden**、**--available**) を追加することで、結果をフィルターにかけることができることに注意してください。利用可能なオプションの詳細は、man ページを参照してください。

- 特定のグループに含まれている必須および任意のパッケージを一覧表示するには、次のコマンドを実行します。

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

`group-name` は、グループ名に置き換えます。

glob 表現を引数として追加して結果をフィルターできることに注意してください。詳細は、[dnf input](#) での [glob 表現の指定](#) を参照してください。

4.6. 利用可能なモジュールの一覧表示

次の手順では、使用可能なモジュールを見つける方法と、**dnf** を使用してそれらの詳細を確認する方法について説明します。

手順

- システムで利用可能なモジュールストリームを一覧表示するには、以下を行います。

```
# dnf module list
```

このコマンドの出力には、モジュールストリームの名前、ストリーム、プロファイル、および概要が1行ごとに表示されます。



注記

モジュールは、今後の RHEL 9 のマイナーリリースで利用できます。

関連情報

- [Modules](#)
- [モジュールストリーム](#)
- [モジュールプロファイル](#)

4.7. DNF INPUT での GLOB 表現の指定

dnf コマンドを使用すると、1つ以上の **glob 表現** を引数として追加することで、結果をフィルタリングできます。glob 表現は、**dnf** コマンドに引数として指定すると、エスケープする必要があります。

以下の手順では、glob 表現が意図した通りに**dnf**に渡されるようにするための2つの方法を説明します。

手順

- glob 表現全体をダブルクォートまたはシングルクォートします。

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

file-name は、ファイルの名前に置き換えます。

file-nameの前には、/または*/の文字シーケンスを置く必要があります。これにより、望ましい結果が得られます。

- ワイルドカード文字の前にはバックスラッシュ (\) を追加して、ワイルドカード文字をエスケープします。

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

file-name は、ファイルの名前に置き換えます。

4.8. 関連情報

- [RHEL 9 でコンテンツを一覧表示するコマンド](#)

第5章 RHEL 9 コンテンツのインストール

次のセクションでは、Red Hat Enterprise Linux 9 にコンテンツをインストールする方法を説明します。

- [セクション 5.1. 「パッケージのインストール」](#) には、パッケージのインストール手順が記載されています。
- [セクション 5.2. 「パッケージグループのインストール」](#) では、パッケージグループのインストール方法を説明します。
- [セクション 5.3. 「インストール済みコンテンツの実行」](#) では、RHEL 9 にインストールされているコンテンツの実行方法を説明します。

5.1. パッケージのインストール

以下の手順では、**dnf** を使用してパッケージをインストールする方法を説明します。

手順

- パッケージのインストール

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

`package-name` を、パッケージ名に置き換えます。

- 複数のパッケージとその依存関係を同時にインストールするには、以下を使用します。

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

`package-name-1` および `package-name-2` は、パッケージ名に置き換えます。

- **multilib** システムにパッケージをインストールする場合に (AMD64、Intel 64 マシン)、パッケージのアーキテクチャーをパッケージ名に追加して指定できます。

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

`package-name.arch` は、パッケージの名前およびアーキテクチャーに置き換えます。

- インストールするバイナリー名は分かっているが、パッケージ名が分からない場合は、引数としてバイナリーへのパスを使用できます。

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

`/usr/sbin/binary-file` は、バイナリーファイルへのパスに置き換えます。

dnf はパッケージリストを検索して、`/usr/sbin/binary-file` を提供するパッケージを見つけ、それをインストールするかどうかを尋ねてきます。

- ローカルディレクトリーからダウンロード済みのパッケージをインストールするには、以下を使用します。

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

`/path/` は、パッケージへのパスに置き換えます。

5.2. パッケージグループのインストール

以下の手順では、**dnf** を使用して、グループ名または groupID でパッケージグループをインストールする方法を説明します。

手順

- パッケージグループをグループ名でインストールするには、以下を使用します。

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

group-name は、グループまたは環境グループのフルネームに置き換えます。

- groupID でパッケージグループをインストールするには、以下のコマンドを実行します。

```
# dnf group install groupID
```

groupID は、グループの ID に置き換えます。

5.3. インストール済みコンテンツの実行

新しいコマンドは、通常、RHEL 9 リポジトリからコンテンツをインストールした後に有効になります。コマンドがRPMパッケージに由来するものであったり、RPMパッケージがモジュールによって有効化されたものであれば、コマンドを使用するエクスペリエンスは変わらないはずです。

手順

- 新しいコマンドを実行するには、次のように実行します。

```
$ command
```

5.4. 関連情報

- **dnf(8)** の man ページ
- [RHEL 9 にコンテンツをインストールするコマンド](#)

第6章 RHEL 9 コンテンツの更新

DNF では、システムに保留中の更新があるかどうかを確認できます。更新が必要なパッケージを一覧表示して、1つのパッケージ、複数のパッケージ、またはすべてのパッケージを一度に更新できます。更新パッケージに依存関係がある場合は、合わせて更新されます。

次のセクションでは、DNF を使用して、Red Hat Enterprise Linux 9 でコンテンツを更新する方法を説明します。

- [セクション 6.1.「更新の確認」](#) では、利用可能な更新を確認する方法を説明します。
- [セクション 6.2.「パッケージの更新」](#) では、1つのパッケージ、パッケージグループ、またはすべてのパッケージとその依存関係を更新する方法を説明します。
- [セクション 6.3.「セキュリティ関連のパッケージの更新」](#) では、セキュリティ更新を適用する方法を説明します。

6.1. 更新の確認

以下の手順では、dnf を使用して、システムにインストールされているパッケージで利用可能な更新を確認する方法を説明します。

手順

- 以下のコマンドを実行して、お使いのシステムに利用可能な更新があるパッケージを確認します。

```
# dnf check-update
```

このコマンドは、更新が利用可能なパッケージおよびその依存関係の一覧を表示します。

6.2. パッケージの更新

以下の手順では、dnf を使用して、1つのパッケージ、パッケージグループ、またはすべてのパッケージとその依存関係を更新する方法を説明します。

手順

- すべてのパッケージとその依存関係を更新するには、次のコマンドを実行します。

```
# dnf upgrade
```

- 1つのパッケージを更新するには、以下のコマンドを実行します。

```
# dnf upgrade package-name
```

package-name を、パッケージ名に置き換えます。

- パッケージグループを更新するには、以下を使用します。

```
# dnf group upgrade group-name
```

group-name は、パッケージグループの名前に置き換えます。



重要

カーネルの更新を適用する際に、dnf は、**dnf upgrade** コマンドまたは **dnf install** コマンドを使用しているかどうかに関わらず、新しいカーネルを常にインストールします。

6.3. セキュリティー関連パッケージの更新

以下の手順では、**dnf** を使用して、セキュリティー関連のパッケージを更新する方法を説明します。

手順

- セキュリティーエラータが含まれる、利用可能な最新パッケージにアップグレードするには以下を使用します。

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

- 最新のセキュリティーエラータパッケージにアップグレードするには、以下を使用します。

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


第7章 RHEL 9 でのソフトウェア更新の自動化

パッケージの更新を自動的にかつ定期的に確認してダウンロードするには、**dnf-automatic** パッケージに含まれる **DNF Automatic** ツールを使用できます。

DNF Automatic は、**DNF** に対する代替のコマンドラインインターフェースで、**systemd** タイマーや **cron** ジョブなどのツールを使用した自動実行や定期実行に適しています。

DNF Automatic は、必要に応じてパッケージメタデータを同期し、利用可能な更新を確認してから、ツールの設定方法に応じて以下のアクションのいずれかを実行します。

- 終了
- 更新済みパッケージのダウンロード
- 更新のダウンロードおよび適用

その後、標準出力やメールなど、選択したメカニズムによって操作の結果が報告されます。

次のセクションでは、Red Hat Enterprise Linux 9 でソフトウェア更新を自動化する方法を説明します。

- [セクション 7.1 「DNF Automatic のインストール」](#) では、**DNF Automatic** ツールのインストール方法を説明します。
- [セクション 7.2 「DNF Automatic 設定ファイル」](#) では、**DNF Automatic** 設定ファイルとそのセクションを説明します。
- [セクション 7.3 「DNF Automatic の有効化」](#) では、**DNF Automatic** ツールを有効にする方法を説明します。
- [セクション 7.4 「dnf-automatic パッケージに含まれる systemd タイマーユニットの概要」](#) では、**dnf-automatic** の **systemd** タイマーユニットが一覧表示されます。

7.1. DNF AUTOMATIC のインストール

以下の手順では、**DNF Automatic** のインストール方法を説明します。

手順

- **dnf-automatic** パッケージをインストールします。

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

検証

- **dnf-automatic** パッケージが存在することを確認して、インストールが正常に完了したことを確認します。

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

7.2. DNF AUTOMATIC 設定ファイル

初期設定では、**DNF Automatic** は **/etc/dnf/automatic.conf** を設定ファイルとして使用し、動作を定義します。

設定ファイルは、以下のトピックセクションに分かれています。

- **[commands]** セクション
DNF Automatic の操作モードを設定します。
- **[emitters]** セクション
DNF Automatic の結果が報告される方法を定義します。
- **[command_email]** セクション
電子メールの送信に使用する外部コマンドのメールエミッター設定を提供します。
- **[email]** セクション
電子メールエミッターの設定を提供します。
- **[base]** セクション
DNF のメイン設定ファイルの設定を上書きします。

`/etc/dnf/automatic.conf` ファイルのデフォルト設定では、DNF Automatic は利用可能な更新を確認し、ダウンロードして、標準出力として結果を報告します。



警告

[commands] セクションの操作モードの設定は、`dnf-automatic.timer` 以外のすべてのタイマーユニットに対して、`systemd` タイマーユニットで使用される設定によって上書きされます。

関連情報

- [DNF Automatic ドキュメント](#)
- `man` ページの `man dnf-automatic`
- `dnf-automatic` パッケージに含まれる `systemd` タイマーユニットの概要

7.3. DNF AUTOMATIC の有効化

DNF Automatic を実行するには、常に特定の `systemd` タイマーユニットを有効にして起動する必要があります。`dnf-automatic` パッケージで提供されるタイマーユニットのいずれかを使用するか、必要に応じて独自のタイマーユニットを作成することができます。

以下の手順では、DNF Automatic を有効にする方法を説明します。

前提条件

- `/etc/dnf/automatic.conf` 設定ファイルを変更して、DNF Automatic の挙動を指定している。

手順

- 利用可能な更新をダウンロードする `systemd` タイマーユニットを選択、有効化、および起動するには、以下のコマンドを実行します。

-

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

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

- 利用可能な更新をダウンロードとインストールする systemd タイマーユニットを選択、有効化、および起動するには、以下のコマンドを実行します。

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

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

- 利用可能な更新をレポートする systemd タイマーユニットを選択、有効化、および起動するには、以下のコマンドを実行します。

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

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

- ダウンロード、ダウンロード、インストール、または利用可能な更新の報告を行う systemd タイマーユニットを選択、有効化、および起動するには、次を使用します。

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

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

- 必要に応じて、**--now** オプションを使用して、1つのコマンドで systemd タイマーユニットを選択、有効に、および起動します。以下に例を示します。

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



注記

コマンドラインまたはカスタムスクリプトから **/usr/bin/dnf-automatic** ファイルを直接実行して、DNF Automatic を実行することもできます。

検証

- タイマーが有効化されていることを確認します。

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

関連情報

- man ページの **man dnf-automatic**
- [dnf-automatic パッケージに含まれる systemd タイマーユニットの概要](#)
- [DNF Automatic 設定ファイル](#)

7.4. DNF-AUTOMATIC に含まれる SYSTEMD タイマーユニットの概要

更新のダウンロードおよび適用時に、systemd タイマーユニットが優先され、`/etc/dnf/automatic.conf` 設定ファイルの設定が上書きされます。

たとえば、以下を設定します。

```
download_updates = yes
```

`/etc/dnf/automatic.conf` 設定ファイルで `dnf-automatic-notifyonly.timer unit` を有効にしても、パッケージはダウンロードされません。

`dnf-automatic` パッケージには、次の systemd タイマーユニットが含まれます。

表7.1 `dnf-automatic` に含まれる systemd タイマー

| タイマーユニット | 機能 | <code>/etc/dnf/automatic.conf</code> ファイルの設定を上書きしますか？ |
|---|---|---|
| <code>dnf-automatic-download.timer</code> | <p>キャッシュにパッケージをダウンロードし、更新を利用できるようにします。</p> <p>注記:このタイマーユニットでは更新パッケージはインストールされません。インストールを実行するには、<code>dnf update</code> コマンドを実行する必要があります。</p> | はい |
| <code>dnf-automatic-install.timer</code> | 更新したパッケージをダウンロードしてインストールします。 | はい |
| <code>dnf-automatic-notifyonly.timer</code> | <p>リポジトリデータのみをダウンロードして、リポジトリキャッシュを最新の状態に保ち、利用可能な更新を通知します。</p> <p>注記:このタイマーユニットでは、更新されたパッケージはダウンロードまたはインストールされません。</p> | はい |
| <code>dnf-automatic.timer</code> | <p>更新のダウンロードおよび適用時のこのタイマーの動作は、<code>/etc/dnf/automatic.conf</code> 設定ファイルの設定により指定されます。</p> <p>デフォルトの挙動は、<code>dnf-automatic-download.timer</code> ユニットと同じで、パッケージをダウンロードしますが、インストールは行いません。</p> | いいえ |

関連情報

- man ページの **man dnf-automatic**
- [DNF Automatic 設定ファイル](#)

第8章 RHEL 9 コンテンツの削除

次のセクションでは、Red Hat Enterprise Linux 9 でコンテンツを削除する方法を説明します。

- [セクション 8.1. 「インストール済みパッケージの削除」](#) では、パッケージの削除について説明します。
- [セクション 8.2. 「パッケージグループの削除」](#) では、パッケージグループの削除について説明します。

8.1. インストール済みパッケージの削除

以下の手順では、`dnf` を使用してパッケージを削除する方法を説明します。

手順

- 特定のパッケージと、使用されていない依存パッケージをすべて削除するには、以下のコマンドを実行します。

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

`package-name` を、パッケージ名に置き換えます。

パッケージは、その他の依存パッケージとともに削除されることに注意してください。

- 複数のパッケージと、その未使用の依存関係を同時に削除するには、以下のコマンドを実行します。

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

`package-name-1` および `package-name-2` は、パッケージ名に置き換えます。



注記

依存パッケージを削除しないと、`dnf` はパッケージを削除できません。

8.2. パッケージグループの削除

以下の手順では、グループ名または `groupID` を使用してパッケージを削除する方法を説明します。

手順

- グループ名を使用してパッケージグループを削除するには、以下を使用します。

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

`group-name` は、グループの完全な名前に置き換えます。

- `groupID` でパッケージグループを削除するには、以下を使用します。

```
# dnf group remove groupID
```

`groupID` は、グループの ID に置き換えます。

8.3. 関連情報

- [RHEL 9 でコンテンツを削除するコマンド](#)

第9章 パッケージ管理履歴の処理

dnf history コマンドを使用すると、以下の情報を確認できます。

- DNF トランザクションのタイムライン
- トランザクションの発生日時
- トランザクションにより影響を受けたパッケージの数
- トランザクションの成功または中止の有無
- トランザクション間で RPM データベースが変更された場合

dnf history コマンドを使用して、トランザクションを元に戻したり、やり直したりすることもできます

本セクションでは、**dnf** を使用して、Red Hat Enterprise Linux 9 でパッケージ管理履歴を処理する方法を説明します。

- [セクション 9.1. 「トランザクションの一覧表示」](#) では、最新のトランザクション、選択したパッケージに対する最新の操作、および特定のトランザクションの詳細を一覧表示する方法を説明します。
- [セクション 9.2. 「トランザクションの復元」](#) では、選択したトランザクションまたは最後のトランザクションを元に戻す方法を説明します。
- [セクション 9.3. 「トランザクションの繰り返し」](#) では、選択したトランザクションまたは最後のトランザクションを繰り返す方法を説明します。

9.1. トランザクションの一覧表示

以下の手順では、最新の DNF トランザクション、選択したパッケージに対する最新の操作、および特定トランザクションの詳細を一覧表示する方法を説明します。

手順

- 最新の DNF トランザクションの一覧を表示するには、以下のコマンドを実行します。

```
# dnf history
```

- 選択したパッケージの最新操作の一覧を表示するには、以下を使用します。

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

`package-name` を、パッケージ名に置き換えます。glob 表現を追加して、コマンドの出力をフィルタリングできます。詳細は、[dnf input](#) での [glob 表現の指定](#) を参照してください。

- 特定のトランザクションの詳細を表示するには、以下のコマンドを実行します。

```
# dnf history info transactionID
```

`transactionID` は、トランザクションの ID に置き換えます。

9.2. トランザクションの復元

以下の手順では、**dnf** を使用して、選択したトランザクションまたは最後のトランザクションを元に戻す方法を説明します。

手順

- 特定のトランザクションを元に戻すには、以下を使用します。

```
# dnf history undo transactionID
```

transactionID は、トランザクションの ID に置き換えます。

- 最後のトランザクションを元に戻すには、以下を使用します。

```
# dnf history undo last
```

dnf history undo コマンドは、トランザクション中に実行した手順のみを元に戻すことに注意してください。このトランザクションで新しいパッケージがインストールされた場合は、**dnf history undo** がそのパッケージをアンインストールします。トランザクションでパッケージがアンインストールされた場合は、**dnf history undo** が再インストールします。また、**dnf history undo** コマンドは、古いパッケージが依然として利用可能な場合、更新されたすべてのパッケージを以前のバージョンにダウングレードしようとします。

9.3. トランザクションの繰り返し

以下の手順では、**dnf** を使用して、選択したトランザクションまたは最後のトランザクションを繰り返す方法を説明します。

手順

- 特定のトランザクションを繰り返すには、以下を使用します。

```
# dnf history redo transactionID
```

transactionID は、トランザクションの ID に置き換えます。

- 最後のトランザクションを繰り返すには、以下を使用します。

```
# dnf history redo last
```

dnf history redo コマンドは、トランザクション中に実行した手順のみを繰り返すことに注意してください。

第10章 カスタムソフトウェアリポジトリの管理

DNF および関連ユーティリティーの設定情報は、`/etc/dnf/dnf.conf` ファイルに保存されます。このファイルには、リポジトリ固有のオプションを設定できる、1つ以上の**[repository]**セクションが含まれます。

`/etc/yum.repos.d/` ディレクトリーの新規または既存の**.repo**ファイルに、個別のリポジトリを定義することが推奨されます。

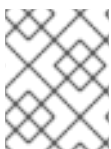
`/etc/dnf/dnf.conf` ファイルの各 **[repository]** セクションで定義した値は、**[main]** セクションで設定した値を上書きすることに注意してください。

次のセクションでは、Red Hat Enterprise Linux 9 でカスタムソフトウェアリポジトリを管理する方法を説明します。

- [セクション 10.1 「DNF リポジトリオプションの設定」](#) では、**[repository]** オプションの設定方法を説明します。
- [セクション 10.2 「DNF リポジトリの追加」](#) では、新しい DNF リポジトリを定義する方法を説明します。
- [セクション 10.3 「DNF リポジトリの有効化」](#) では、システムに追加した DNF リポジトリを有効にする方法を説明します。
- [セクション 10.4 「DNF リポジトリの無効化」](#) では、システムに追加した DNF リポジトリを無効にする方法を説明します。

10.1. DNF リポジトリオプションの設定

`/etc/dnf/dnf.conf` 設定ファイルには、**[repository]** セクションが含まれます。リポジトリ は一意のリポジトリ ID です。**[repository]** セクションでは、個別の DNF リポジトリを定義できます。



注記

競合を回避するために、カスタムリポジトリには Red Hat リポジトリで使用される名前を指定しないでください。

利用可能な **[repository]** オプションの完全なリストは、`dnf.conf(5)` man ページの **[repository] OPTIONS** セクションを参照してください。

10.2. DNF リポジトリの追加

新しいリポジトリを定義するには、以下のいずれかの方法を使用します。

- **[repository]** セクションを `/etc/dnf/dnf.conf` ファイルに追加します。
- `/etc/yum.repos.d/` ディレクトリーの **.repo** ファイルに **[repository]** セクションを追加します。インストールされた RPM またはソフトウェア管理ツール（Subscription Manager など）は、独自の **.repo** ファイルを提供できます。



注記

このディレクトリーの **.repo** ファイル拡張子が付いた全ファイルとして、`/etc/ dnf /dnf.conf` ではなく、**.repo** ファイルにリポジトリを定義します。

以下の手順では、DNF リポジトリをシステムに追加する方法を説明します。

手順

- システムにリポジトリを追加します。

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

repository_url は、リポジトリを参照する URL に置き換えます。



警告

ソフトウェアパッケージを、Red Hat の認証ベース **Content Delivery Network (CDN)** 以外の未検証または信頼できないソースから取得してインストールする場合には、セキュリティ上のリスクが伴います。セキュリティ、安定性、互換性、保全性に関する問題につながる恐れがあります。

10.3. DNF リポジトリの有効化

以下の手順では、システムに追加した DNF リポジトリを有効にする方法を説明します。

手順

- リポジトリを有効にします。

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

repositoryID は、一意のリポジトリ ID に置き換えます。

関連情報

[リポジトリの一覧表示](#)

10.4. DNF リポジトリの無効化

以下の手順では、システムに追加した DNF リポジトリを無効にする方法を説明します。

手順

- リポジトリを無効にします。

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

repositoryID は、一意のリポジトリ ID に置き換えます。

関連情報

[リポジトリの一覧表示](#)

付録A DNF コマンド一覧

本章では、Red Hat Enterprise Linux 9 でコンテンツの一覧表示、インストール、および削除を行うための DNF コマンドを説明します。

A.1. RHEL 9 でコンテンツを一覧表示するコマンド

次の表は、RHEL 9 でコンテンツとその詳細を検索する際に一般的に使用される DNF コマンドを記載しています。

| コマンド | 詳細 |
|--|--|
| dnf search term | パッケージに関連する用語を使ってパッケージを検索する |
| dnf repoquery package | 選択したパッケージで利用可能な DNF リポジトリを検索する |
| dnf list | インストールされているすべてのパッケージと、利用可能なパッケージの一覧を表示します。 |
| dnf list --installed dnf repoquery --installed | システムにインストールされているパッケージの一覧を表示します。 |
| dnf list --available dnf repoquery | インストール可能な有効なリポジトリにある、すべてのパッケージの一覧を表示します。 |
| dnf repolist | システムで有効なリポジトリの一覧を表示します。 |
| dnf repolist --disabled | システムで無効になっているリポジトリの一覧を表示します。 |
| dnf repolist --all | 有効および無効の両方のリポジトリの一覧を表示します。 |
| dnf repoinfo | リポジトリの追加情報の一覧を表示します。 |
| dnf info package-name dnf repoquery --info package_name | 利用可能なパッケージの詳細の表示 |
| dnf repoquery --info --installed package_name | システムにインストールされているパッケージの詳細を表示します。 |

| コマンド | 詳細 |
|----------------------------------|--|
| dnf module list | モジュールとその現在の状態を一覧表示します。 なお、パッケージがどのモジュールからも利用できない場合は、このコマンドの出力は空になります。 |
| dnf group summary | インストールされているグループと利用可能なグループの数を表示する |
| dnf group list | インストール済みおよび利用可能なグループの一覧を表示します。 |
| dnf group info group-name | 特定のグループに含まれる必須および任意のパッケージを一覧表示します。 |

A.2. RHEL 9 にコンテンツをインストールするコマンド

以下の表は、RHEL 9 にコンテンツをインストールするために一般的に使用される DNF コマンドの一覧です。

| コマンド | 詳細 |
|--|---|
| dnf install package-name | パッケージのインストール パッケージがモジュールストリームにより提供されている場合は、 dnf が必要なモジュールストリームを解決し、このパッケージのインストール時に自動的に有効にします。これは、すべてのパッケージ依存関係で再帰的に行われます。より多くのモジュールストリームがこの要件を満たす必要がある場合は、デフォルトのモジュールストリームが使用されます。 |
| dnf install package-name-1 package-name-2 | 複数のパッケージとその依存関係の同時インストール |
| dnf install package-name.arch | multilib システムにパッケージをインストールする際に、パッケージのアーキテクチャーを指定するには、パッケージ名に追加します (AMD64、Intel 64 マシン)。 |
| dnf install /usr/sbin/binary-file | バイナリーへのパスを引数として使用してバイナリーをインストールします。 |
| dnf install /path/ | ダウンロード済みのパッケージをローカルディレクトリーからインストールします。 |

| コマンド | 詳細 |
|-------------------------------------|---|
| dnf install package-url | <p>パッケージ URL を使用したりリモートパッケージのインストール</p> <p>このモジュールは、パッケージをシステムで利用できるようにしたいが、現在はいずれもインストールしたくない場合に有効にします。</p> <p>モジュールによっては、デフォルトのストリームを定義しない場合があります。このような場合は、ストリームを明示的に指定する必要があります。</p> <p>モジュールがデフォルトのストリームを定義する場合は、ストリームとコロンを省略できます。</p> <p>このコマンドを実行しても RPM パッケージはインストールされないことに注意してください。</p> <p>モジュールによってはデフォルトのストリームを定義しないことに注意してください。</p> <p>このコマンドを実行すると、指定したストリームも有効になっていることに注意してください。</p> |
| dnf group install group-name | グループ名でパッケージグループをインストールする |
| dnf group install groupID | groupID でパッケージグループをインストールします。 |

A.3. RHEL 9 でコンテンツを削除するコマンド

以下の表は、RHEL 9 で内容を削除するために一般的に使用される DNF コマンドを記載しています。

| コマンド | 詳細 |
|---|------------------------------|
| dnf remove package-name | 特定のパッケージおよびすべての依存パッケージの削除 |
| dnf remove package-name-1 package-name-2 | 複数のパッケージと、その未使用の依存関係を同時に削除する |
| dnf group remove group-name | グループ名でパッケージグループを削除する |

| コマンド | 詳細 |
|---------------------------------|---|
| dnf group remove groupID | <p>groupID でパッケージグループを削除する</p> <p>このコマンドを実行すると、システムから重要なパッケージが削除される可能性があることに注意してください。</p> <p>このコマンドを実行しても、指定したモジュールからパッケージが削除されないことに注意してください。</p> <p>このコマンドを実行しても、指定したモジュールからパッケージが削除されないことに注意してください。</p> |