



Red Hat Enterprise Linux 8

ユーザー領域コンポーネントのインストール、管理、および削除

Red Hat Enterprise Linux 8 の AppStream および BaseOS

Red Hat Enterprise Linux 8 ユーザー領域コンポーネントのインストール、管理、および削除

Red Hat Enterprise Linux 8 の AppStream および BaseOS

法律上の通知

Copyright © 2019 Red Hat, Inc.

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

本書では、Red Hat Enterprise Linux 8 の AppStream リポジトリおよび BaseOS リポジトリでコンテンツを検索、検出、インストール、および使用方法を説明します。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

RED HAT ドキュメントへのフィードバック	3
第1章 APPSTREAM の使用	4
1.1. RHEL 8 のコンテンツの配布	4
1.2. アプリケーションストリーム	4
1.3. RHEL 8 のパッケージング方法	5
1.4. RHEL 8 で YUM を使用したパッケージ管理	5
第2章 モジュールの概要	6
2.1. モジュールストリーム	6
2.2. モジュールプロファイル	7
第3章 RHEL 8 コンテンツの検索	8
3.1. パッケージの検索	8
3.2. 利用可能なモジュールの一覧表示	8
3.3. 例: モジュールに関する詳細の検索	9
3.4. コンテンツを一覧表示するコマンド	11
第4章 RHEL 8 コンテンツのインストール	13
4.1. パッケージのインストール	13
4.2. パッケージのインストール前にストリームの選択	13
4.3. モジュールストリームのインストール	14
4.4. 例: アプリケーションのデフォルト以外のストリームのインストール	14
4.5. インストール済みコンテンツの実行	15
4.6. RHEL 8 コンテンツのインストールコマンド	16
4.7. 関連資料	17
第5章 RHEL 8 コンテンツの削除	18
5.1. インストール済みパッケージの削除	18
5.2. インストール済みモジュールの削除	18
5.3. モジュールストリームのリセット	19
5.4. コンテンツを削除するコマンド	19
第6章 APPLICATION STREAM コンテンツのバージョン管理	20
6.1. モジュールの依存関係とストリームの変更	20
6.2. モジュールおよび非モジュール依存関係のやり取り	21
6.3. モジュールストリームのリセット	21
6.4. モジュールのストリームをすべて無効化	21
6.5. 異なるバージョンのコンテンツをインストールするモジュールストリームの切り替え	22

RED HAT ドキュメントへのフィードバック

ドキュメントの改善に関するご意見やご要望をお聞かせください。

- 特定の文章に簡単なコメントを記入する場合は、ドキュメントが Multi-page HTML 形式になっているのを確認してください。コメントを追加する部分を強調表示し、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、表示された手順に従ってください。
- より詳細なフィードバックを行う場合は、Bugzilla のチケットを作成します。
 1. [Bugzilla](#) の Web サイトにアクセスします。
 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 APPSTREAM の使用

次のセクションでは、Red Hat Enterprise Linux 8 の AppStream リポジトリに関する概念を説明します。

- 「[RHEL 8 のコンテンツの配布](#)」では、Red Hat Enterprise Linux 8 のコンテンツがどのように BaseOS と AppStream に分割されるかを説明します。
- 「[アプリケーションストリーム](#)」では、Application Stream の概要を説明します。
- 「[RHEL 8 のパッケージング方法](#)」では、AppStream が提供するコンテンツの種類を説明します。
- 「[RHEL 8 で YUM を使用したパッケージ管理](#)」では、Red Hat Enterprise Linux 8 で提供される YUM パッケージマネージャーが従来のモジュール機能をどのように組み合わせるかを説明します。

1.1. RHEL 8 のコンテンツの配布

RHEL 8 のコンテンツは、BaseOS と AppStream の 2 つのメインリポジトリにより配布されます。

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 リポジトリのコンテンツは、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、ユーザー領域アプリケーション、ランタイム言語、およびデータベースが同梱されます。AppStream のコンテンツは、従来の RPM 形式と、RPM 形式の拡張 (**モジュール**) の 2 つの形式から選択できます。



重要

基本的な RHEL インストールには、BaseOS と AppStream の両方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。インストール方法は「[RHEL のインストールおよびデプロイメント](#)」を参照してください。

1.2. アプリケーションストリーム

Red Hat Enterprise Linux 8 では、アプリケーションストリームの概念 (ユーザー領域コンポーネントのバージョン) が導入されました。ユーザー領域コンポーネントのバージョンが複数配信され、オペレーティングシステムのコアパッケージよりも頻繁に更新されるようになりました。これによりプラットフォームや特定デプロイメントの基本的な安定性に影響を及ぼすことなく、Red Hat Enterprise Linux をカスタマイズする柔軟性が向上しました。

アプリケーションストリームとして利用できるようになったコンポーネントは、モジュールまたは RPM パッケージとしてパッケージ化でき、Red Hat Enterprise Linux 8 の AppStream リポジトリから配信されます。AppStream コンポーネントにはそれぞれライフサイクルが定められています。



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、AppStream コンポーネントとはみなされません。

各モジュールは、RHEL のライフサイクルではなく、アプリケーションの寿命に合わせて独自のライフサイクルを定義しています。利用可能なアプリケーションストリームと、そのサポート期間は、「[Red Hat Enterprise Linux 8 Application Streams Life Cycle](#)」を参照してください。

関連資料

- [Red Hat Enterprise Linux 8 Application Streams Life Cycle](#)
- [Red Hat Enterprise Linux のライフサイクル](#)

1.3. RHEL 8 のパッケージング方法

AppStream リポジトリには、2つの方法でパッケージ化されたコンテンツが含まれます。

個別の RPM パッケージ

従来の RPM パッケージは、即時にインストールを行うために使用できます。

モジュール

モジュールは、論理ユニット (アプリケーション、言語スタック、データベース、またはツールセット) を表すパッケージの集まりです。これらのパッケージはまとめてビルドされ、テストされ、そしてリリースされます。

1.4. RHEL 8 で YUM を使用したパッケージ管理

現在、YUM パッケージ管理ツールは DNF 技術に基づいており、新しいモジュール機能に対応しています。

個々の RPM パッケージを扱う YUM の仕様は変更していません。モジュールコンテンツに対応する **yum module** コマンドが追加されています。詳細は [4章 RHEL 8 コンテンツのインストール](#) を参照してください。

モジュール機能は、必要に応じてモジュールとストリームの適切な組み合わせを自動的に選択し、パッケージの論理セットのインストールを可能にするため便利です。

第2章 モジュールの概要

個別の RPM パッケージの他に、AppStream リポジトリにはモジュールが含まれます。モジュールは、コンポーネントを表し、通常は一緒に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ーを含むパッケージが含まれます。

以降のセクションでは、モジュール内で統合および処理する機能を説明します。

- ストリーム - バージョンによるコンテンツの編成。詳細は「[モジュールストリーム](#)」を参照してください。
- プロファイル - 目的によるコンテンツの編成。詳細は「[モジュールプロファイル](#)」を参照してください。

2.1. モジュールストリーム

モジュールのストリームは、AppStream 物理リポジトリ内の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内の 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、デフォルトとしてマークされている場合、またはユーザーにより明示的に有効になった場合に限り、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。別々のコンテナ内で異なるバージョンを使用することは可能です。

各モジュールには、通常、コンポーネントの最新または推奨されるバージョンを提供するデフォルトのストリームを追加できます。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、その他のモジュールストリームに依存します。たとえば、モジュールストリームの `perl-App-cpanminus`、`perl-DBD-MySQL`、`perl-DBD-Pg`、`perl-DBD-SQLite`、`perl-DBI`、`perl-YAML`、および `freeradius` は、`perl` モジュールストリームに依存します。

例2.1 postgresql モジュールストリーム

`postgresql` モジュールは、ストリーム 9.6 および 10 で、バージョンがそれぞれ 9.6 および 10 の PostgreSQL データベースを提供します。ストリーム 10 は現在デフォルトとなっているストリームです。つまり、`postgresql` と指定すると、システムは `postgresql-10.6` パッケージのインストールを試みます。

関連資料

- モジュールの依存関係の詳細は「[モジュールの依存関係とストリームの変更](#)」を参照してください。

2.2. モジュールプロファイル

プロファイル は、サーバー、クライアント、開発、最小インストールなど、特定のユースケースでまとめてインストールされる推奨パッケージの一覧です。このパッケージ一覧は、モジュールストリームの外にあるパッケージ (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。

ユーザの利便性のため一度で完了する、プロファイルを利用した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1つが **デフォルト** となり、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。

例2.2 httpd モジュールプロファイル

Apache Web サーバーを提供する **httpd** モジュールは、インストールに以下のプロファイルを提供します。

- **common** - 強固な実稼働対応のデプロイメント (デフォルトのプロファイル)
- **devel** - **httpd** を修正するのに必要なパッケージ
- **minimal** - 実行中の Web サーバーを提供するパッケージの最小セット

第3章 RHEL 8 コンテンツの検索

次のセクションでは、Red Hat Enterprise Linux 8 の AppStream リポジトリおよび BaseOS リポジトリでコンテンツを検索して調べる方法を説明します。

- 「[パッケージの検索](#)」では、目的のコンテンツを提供するパッケージの検索方法を説明します。
- 「[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を表示し、その詳細を確認する方法を説明します。
- 「[例: モジュールに関する詳細の検索](#)」では、モジュールの詳細を調べるのに必要な手順を説明します。
- 「[コンテンツを一覧表示するコマンド](#)」では、コンテンツを調べるのに便利なコマンドのリファレンスを提供します。

3.1. 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。

前提条件

- 目的のアプリケーションやコンテンツの名前が分かっている。

手順

1. アプリケーション名などのテキスト文字列でパッケージを検索します。

```
$ yum search "text string"
```

2. パッケージの詳細を表示します。

```
$ yum info package
```

3.2. 利用可能なモジュールの一覧表示

ここでは、利用可能なモジュールと、そのモジュールの詳細を検索する手順を説明します。

手順

1. システムで使用できるモジュールストリームの一覧を表示します。

```
$ yum 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。

2. モジュールに関する詳細が表示されます。これには、説明、プロファイル一覧、および提供されるパッケージ一覧が含まれます。

```
$ yum module info module-name
```

- 必要に応じて、モジュールプロファイルごとに、インストールされているパッケージの一覧を表示することもできます。

```
$ yum module info --profile module-name
```

- モジュールの現在の状態を表示します。これには有効なストリームやインストール済みプロファイルが含まれます。

```
$ yum module list module-name
```

関連資料

- 2章 [モジュールの概要](#)

3.3. 例: モジュールに関する詳細の検索

この例は、AppStream リポジトリでモジュールを見つける方法と、そのコンテンツに関する詳細を探す方法を説明します。



注記

この例では、見やすくするために出力が編集されています。実際の出力には、ここに示されているよりも多くの情報が含まれている場合があります。

手順

- 利用可能なモジュールを一覧表示します。

```
$ yum module list
Name      Stream Profiles  Summary
(...)
postgresql 10 [d] client, PostgreSQL server and client module
           server [d]
postgresql 9.6  client, PostgreSQL server and client module
           server [d]
(...)

Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled
```

- `postgresql` モジュールの詳細を確認します。

```
$ yum module info postgresql

Name       : postgresql
Stream     : 10 [d][a]
Version    : 820190104140132
Context    : 9edba152
Profiles   : client, server [d]
Default profiles : server
Repo       : appstream-internal-nightly
Summary    : PostgreSQL server and client module
Description : (...)
(...)
```

```

Name       : postgresql
Stream     : 9.6
Version    : 820190104140337
Context    : 9edba152
Profiles   : client, server [d]
Default profiles : server
Repo       : appstream-internal-nightly
Summary    : PostgreSQL server and client module
Description : (...)
(...)

Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled, [a]ctive

```

指定されたストリームがないため、すべてのストリームが表示されます。

3. `postgresql` モジュールのストリーム `10` にあるプロファイルを確認します。

```

$ yum module info --profile postgresql:10
(...)
Name   : postgresql:10:820190104140132:9edba152:x86_64
client : postgresql
server : postgresql-server

```

各プロファイルは、依存関係を含む複数のパッケージセットをインストールします。

4. デフォルトのストリーム `10` とプロファイル `server` を使用して `postgresql` モジュールをインストールします。

```

# yum install @postgresql
Dependencies resolved.
=====
Package      Version                Repository Size
=====
Installing group/module packages:
postgresql-server 10.6-1.module+el8+2469+5ecd5aae appstream 5.1 M
Installing dependencies:
libpq         10.5-1.el8             appstream 188 k
postgresql    10.6-1.module+el8+2469+5ecd5aae appstream 1.5 M
Installing module profiles:
postgresql/server
Enabling module streams:
postgresql     10

Transaction Summary
=====
Install 3 Packages

Total download size: 6.7 M
Installed size: 27 M
Is this ok [y/N]: y
(...)

```

ストリーム `10` が有効になり、プロファイル `server` のパッケージがインストールされます。

5. `postgresql` モジュールの現在の状態を確認します。

```
$ yum module list postgresql
Name      Stream  Profiles          Summary
postgresql 10 [d][e] client, server [d] [i] (...)
postgresql 9.6     client, server [d]  (...)
```

Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled

この出力は、デフォルトストリーム 10 が有効になり、そのプロファイル **server** がインストールされていることを表しています。

3.4. コンテンツを一覧表示するコマンド

ここでは、コンテンツとその詳細を検索するのに一般的に使用されるコマンドを取り上げます。

コマンドリスト

利用可能なパッケージの一覧表示

```
$ yum list available
```

任意のテキスト文字列を使用したパッケージの検索

```
$ yum search "text string"
```

パッケージの詳細の表示

```
$ yum info package
```

パッケージを提供するモジュールの検索

```
$ yum module provides package
```

パッケージがモジュール外で利用できる場合に、このコマンドの出力は空になります。

利用可能なモジュールの一覧表示

```
$ yum module list
```

モジュールの詳細の表示

```
$ yum module info module-name
```

デフォルトストリームを使用した、モジュールのプロファイルによりインストールされたパッケージの一覧表示

```
$ yum module info --profile module-name
```

指定のストリームを使用した、モジュールのプロファイルによりインストールされたパッケージの一覧表示

```
$ yum module info --profile module-name:stream
```

モジュールの現在の状態の表示

```
$ yum module list module-name
```


第4章 RHEL 8 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 8 にコンテンツをインストールする方法を説明します。

- 「[パッケージのインストール](#)」では、パッケージのインストール手順を説明します。
- 「[パッケージのインストール前にストリームの選択](#)」では、パッケージインストールのストリームの選択方法を説明します。
- 「[モジュールストリームのインストール](#)」では、モジュールが提供するパッケージセットのインストール手順を説明します。
- 「[例: アプリケーションのデフォルト以外のストリームのインストール](#)」では、デフォルト以外のバージョンのパッケージセットをインストールするのに必要な手順を説明します。
- [???](#)では、RHEL 8 のインストール済みコンテンツの実行に関する詳細を説明します。
- 「[RHEL 8 コンテンツのインストールコマンド](#)」では、RHEL 8 コンテンツをインストールするときに便利なコマンドのリファレンスを紹介します。

4.1. パッケージのインストール

ここでは、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

```
# yum install package
```

- パッケージがモジュールストリームで提供されていない場合、この手順は、Red Hat Enterprise Linux の旧バージョンの手順と同じになります。
- パッケージが、有効になっているモジュールストリームにより提供されている場合、パッケージをインストールするのに必要な追加操作はありません。
- パッケージが、デフォルトのモジュールストリームにより提供されている場合は、このパッケージをインストールする前に、**yum** ツールがそのモジュールストリームを自動的にかつ透過的に有効にします。
- パッケージが、アクティブでない(上述のいずれの例でもない)モジュールストリームにより提供されている場合は、その各モジュールストリームを手動で有効にするまで認識されません。

関連資料

- 「[モジュールストリームのインストール](#)」
- 「[RHEL 8 で YUM を使用したパッケージ管理](#)」

4.2. パッケージのインストール前にストリームの選択

デフォルトのモジュールストリームを使用すると、モジュール機能に関わらずパッケージを確実にインストールします。デフォルト以外のストリームのバージョンのパッケージが必要な場合は、パッケージをインストールする前に、それを提供するストリームを有効にする必要があります。

前提条件

- [アクティブなモジュールストリーム の概念](#) を理解している。

手順

- モジュールストリームを有効にします。

```
# yum module enable module-name:stream
```

module-name および **stream** を、モジュールおよびストリームの名前に置き換えます。

yum が確認を求め、ストリームが有効になり、アクティブになります。モジュールの別のストリームが (デフォルトまたは有効にすることで) アクティブになっていた場合は、アクティブではなくなります。

4.3. モジュールストリームのインストール

ここでは、モジュールストリームを使用して、推奨されるパッケージセットをそのモジュールからインストールする方法を説明します。

前提条件

- [アクティブなモジュールストリーム の概念](#) を理解している。
- 同じモジュールの別のストリームからパッケージをインストールしていない。

手順

- モジュールストリームのプロファイルをインストールします。

```
# yum install @module-name:stream/profile
```

これにより、ストリームが有効になり、モジュールの指定ストリーム (バージョン) およびプロファイル (目的) に対して推奨されるパッケージセットがインストールされます。

デフォルトプロファイルを使用するため、`/profile` は指定しません。この手順では、プロファイルを指定しないと失敗するため、プロファイルをデフォルトで設定していない場合は指定する必要があります。

また、アクティブなストリームを使用するため、`:stream` も指定しません。モジュールにアクティブなストリームが存在しない場合は、ストリームを指定する必要があります。

関連資料

- [2章 モジュールの概要](#)
- [「RHEL 8 コンテンツのインストールコマンド」](#)

4.4. 例: アプリケーションのデフォルト以外のストリームのインストール

この例では、デフォルト以外のストリーム (バージョン) からアプリケーションをインストールする方法を示します。

具体的には、この例ではバージョン 9.6 の PostgreSQL サーバー (パッケージ `postgresql-server`) をインストールします。一方、デフォルトのストリームでは、バージョン 10 が提供されます。

手順

1. `postgresql-server` パッケージを提供するモジュールを一覧表示し、使用できるストリームを確認します。

```
$ yum module list postgresql
Name      Stream Profiles      Summary
postgresql 10 [d] client, default [d] PostgreSQL server and client module
postgresql 9.6  client, default [d] PostgreSQL server and client module

Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled
```

出力は、ストリームが 10 および 9.6 の `postgresql` モジュールが使用できることを表しています。デフォルトのストリームは 10 です。

2. ストリーム 9.6 の `postgresql` モジュールにより提供されるパッケージをインストールします。

```
# yum install @postgresql:9.6
Dependencies resolved.
=====
Package          Version                Repository Size
=====
Installing group/module packages:
postgresql-server 9.6.10-1.module+el8+2470+d1bafa0e appstream 5.0 M
Installing dependencies:
libpq             10.5-1.el8             appstream 188 k
postgresql        9.6.10-1.module+el8+2470+d1bafa0e appstream 1.4 M
Installing module profiles:
postgresql/server
Enabling module streams:
postgresql        9.6

Transaction Summary
=====
Install 3 Packages

Total download size: 6.6 M
Installed size: 27 M
Is this ok [y/N]: y
(...)
Complete!
```

インストールプロファイルが指定されていなかったため、デフォルトのプロファイル `server` が使用されていました。

3. インストールされた PostgreSQL のバージョンを確認します。

```
$ postgres --version
postgres (PostgreSQL) 9.6.10
```

4.5. インストール済みコンテンツの実行

通常は、RHEL 8 リポジトリからコンテンツをインストールすると、新しいコマンドが有効になります。コマンドが RPM パッケージに由来するか、モジュールにより有効になった RPM パッケージに由来するかは、コマンドの使用方法に影響しません。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
```

4.6. RHEL 8 コンテンツのインストールコマンド

ここでは、RHEL 8 コンテンツをインストールするのに一般的に使用されるコマンドを取り上げます。

コマンドリスト

パッケージのインストール

```
# yum install package
```

モジュールストリームによりパッケージが提供される場合は、**yum** 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なります。これは、すべてのパッケージ依存関係にも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依存関係を満たすために必要な場合は、デフォルトのモジュールストリームが使用されます。

モジュールのデフォルトのストリームの有効化

```
# yum module enable module-name
```

システムでパッケージを使用できるようにし、現時点ではインストールしない場合は、モジュールを有効にします。

デフォルトのストリームを定義しないモジュールもあります。そのような場合は、ストリームを明示的に指定する必要があります。

モジュールの特定のストリームの有効化

```
# yum module enable module-name:stream
```

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。

デフォルトのストリームとプロファイルを使用してモジュールのインストール

```
# yum install @module-name
```

または

```
# yum module install module-name
```

注意

デフォルトのストリームを定義しないモジュールもあります。

特定のストリームとデフォルトのプロファイルを使用したモジュールのインストール

```
# yum install @module-name:stream
```

または

```
# yum module install module-name:stream
```

特定のストリームおよびプロファイルを使用したモジュールのインストール

```
# yum install @module-name:stream/profile
```

または

```
# yum module install module-name:stream/profile
```

4.7. 関連資料

オンラインリソース

- 従来のソフトウェアのインストール方法の詳細は、『[基本的なシステム設定の構成](#)』の「[yumを使用したソフトウェアのインストール](#)」を参照してください。

インストール済みのリソース

- さまざまな **yum** ツールコマンドの詳細は、man ページの **yum(8)** を参照してください。

```
$ man yum
```

第5章 RHEL 8 コンテンツの削除

次のセクションでは、Red Hat Enterprise Linux 8 でコンテンツを削除する方法を説明します。

- 「[インストール済みパッケージの削除](#)」では、パッケージの削除を説明します。
- 「[インストール済みモジュールの削除](#)」では、モジュールストリームからインストールされたコンテンツを削除する方法を説明します。
- 「[モジュールストリームのリセット](#)」では、モジュールストリームを初期状態にリセットする方法を説明します。
- 「[コンテンツを削除するコマンド](#)」は、コンテンツを削除するためのコマンドをまとめたものです。

5.1. インストール済みパッケージの削除

本セクションは、パラメーターを削除する方法を説明します。

手順

- パッケージを削除します。

```
# yum remove package
```

パッケージは、その他の依存関係パッケージとともに削除されます。

5.2. インストール済みモジュールの削除

モジュールを削除すると、現在有効なモジュールストリームのプロファイルからインストールしたパッケージがすべて削除され、これらのパッケージに依存するパッケージやモジュールも削除されます。

このモジュールストリームからインストールされ、プロファイルに記載されていないパッケージは、システムにインストールされたままとなり、手動で削除できます。

前提条件

- 削除するモジュールに、インストール済みプロファイルがある。
- [モジュール依存関係の解決](#) を理解している。

手順

1. モジュールからインストールしたパッケージの一覧を表示します。

```
$ yum module info module-name | grep module+el8 | sed 's/.*: //g;s/\n/ /g' | xargs yum list installed
```

module-name 名を、モジュール名に置き換えます。これにより、このモジュールからインストールされたパッケージの一覧が表示されます。

2. 前の手順で一覧表示したパッケージを削除します。

```
# yum remove package
```

-

package を、前の手順に記載されているパッケージに置き換えます。パッケージが複数になる場合は、その名前をスペースで区切ります。**yum** ツールは変更の概要を提示し、確認を求めます。

3. モジュールプロファイルをアンインストールとしてマークします。

```
# yum module remove module-name
```

現在有効なモジュールストリームは有効のままになります。

4. 必要に応じて、ストリームをリセットまたは無効にします。

5.3. モジュールストリームのリセット

モジュールをリセットすると、そのストリームをすべて (有効でも無効でもない) 初期状態にリセットします。モジュールにデフォルトストリームがある場合は、モジュールをリセットするとそのストリームがアクティブになります。

手順

- モジュールストリームをリセットにします。

```
# yum module reset module-name
```

モジュールのすべてのストリームが初期状態に戻ります。インストールしたコンテンツは削除されません。

5.4. コンテンツを削除するコマンド

ここでは、コンテンツを削除するのに一般的に使用されるコマンドを取り上げます。

コマンドリスト

パッケージの削除

```
# yum remove package
```

インストール済みモジュールストリームのプロファイルの削除

```
# yum module remove module-name
```

モジュールのストリームをすべて初期状態にリセット

```
# yum module reset module-name
```

モジュールおよびそのストリームの無効化

```
# yum module disable module-name
```

第6章 APPLICATION STREAM コンテンツのバージョン管理

AppStream リポジトリのコンテンツは、モジュールストリームに対応する複数のバージョンが使用できます。本章では、新しいモジュールストリームを有効にする以外の、有効なモジュールストリームに対して変更する際に必要な操作を説明します。

- 「[モジュールの依存関係とストリームの変更](#)」では、モジュール依存関係ルールを説明します。
- 「[モジュールおよび非モジュール依存関係のやり取り](#)」では、モジュールストリームの依存関係が、パッケージの依存関係の処理にどのように影響するかを詳細に説明します。
- 「[モジュールストリームのリセット](#)」では、モジュールを初期状態にリセットする手順を説明します。
- 「[モジュールのストリームをすべて無効化](#)」では、モジュール、およびそのストリームを完全に無効にする手順を説明します。
- 「[異なるバージョンのコンテンツをインストールするモジュールストリームの切り替え](#)」では、一部のコンテンツがすでにインストールされている場合に、モジュールのアクティブなストリームを切り替えることで、別バージョンのコンテンツをインストールする方法を説明します。

6.1. モジュールの依存関係とストリームの変更

これまで、コンテンツを提供するパッケージは他のパッケージに依存し、通常は希望の依存関係バージョンを指定していました。モジュールに含まれるパッケージにもこの仕組みが適用されますが、パッケージとその特定バージョンをモジュールとストリームとしてグループ化したことで、さらに制限されます。また、モジュールストリームは含まれるパッケージや提供するパッケージに制限されずに、他のモジュールのストリームへの依存関係を宣言できます。

パッケージやモジュールの操作後、インストール済みの全パッケージの依存関係ツリー全体が、パッケージが宣言する状態と適合する必要があります。また、すべてのモジュールストリームの依存関係に適合する必要があります。

その結果、以下のことが言えます。

- モジュールストリームを有効にすると、他のモジュールのストリームを有効にすることが必要になる場合があります。
- モジュールストリームプロファイルのインストールまたはストリームからのパッケージのインストールには、他のモジュールのストリームを有効化したり、他のパッケージをインストールすることが必要になる場合があります。
- モジュールのストリームを無効にするには、他のモジュールストリームを無効にする必要があります。自動的に削除されるパッケージはありません。
- パッケージの削除には、他のパッケージの削除が必要になることがあります。このようなパッケージがモジュールにより提供されている場合は、このモジュールストリームのパッケージがインストールされなくなっても、将来のインストールに備えてモジュールストリームは有効のままになります。これは、未使用の yum リポジトリの挙動を反映しています。

同じモジュールの別のストリームが有効になっている場合は、モジュールのストリームを有効にすることができません。ストリームを切り替えるには、モジュールをリセットしてから、新しいストリームを有効にする必要があります。別のストリームに切り替える前に、ストリームからインストールしたパッ

ページをすべて削除することが推奨されます。これは、対応するリポジトリやストリームがないのにパッケージがインストールされている状態になるのを防ぐためです。

技術的には、モジュールをリセットしても、インストール済みパッケージを自動的に変更しません。以前のストリームが提供するパッケージと、そのパッケージに依存するパッケージを削除することは、手動で行う必要があります。

6.2. モジュールおよび非モジュール依存関係のやり取り

モジュール依存関係 は、通常の RPM 依存関係に追加された層になります。モジュール依存関係は、リポジトリ間で仮想的な依存関係と同様に動作します。これは、異なるパッケージをインストールすると、RPM の依存関係の解決に加え、モジュールの依存関係が事前に解決されている必要があることを意味します。

明示的に変更が指示される場合を除き、システムは常にモジュールとストリームの選択を保持します。モジュールパッケージは、このパッケージを提供するモジュールの、現在有効なストリームに含まれる更新を受け取りますが、別のストリームに含まれたバージョンへはアップグレードしません。

6.3. モジュールストリームのリセット

モジュールをリセットすると、そのストリームをすべて (有効でも無効でもない) 初期状態にリセットします。モジュールにデフォルトストリームがある場合は、モジュールをリセットするとそのストリームがアクティブになります。

手順

- モジュールストリームをリセットにします。

```
# yum module reset module-name
```

モジュールのすべてのストリームが初期状態に戻ります。インストールしたコンテンツは削除されません。

6.4. モジュールのストリームをすべて無効化

デフォルトストリームがあるモジュールには、常にアクティブなストリームが1つ含まれます。すべてのモジュールストリームからそのコンテンツにアクセスすべきでない状況では、モジュール全体を無効にできます。

前提条件

- [アクティブなモジュールストリームの概念](#) を理解している。

手順

- モジュールを無効にします。

```
# yum module disable module-name
```

yum が確認を求め、その後、そのすべてのストリームを持つモジュールを無効にします。すべてのモジュールが非アクティブになります。インストール済みのコンテンツは削除されません。

6.5. 異なるバージョンのコンテンツをインストールするモジュールストリームの切り替え

別のモジュールストリームへの切り替えは、通常、システムに現在インストールされているバージョン以外のバージョンに、パッケージをアップグレードまたはダウングレードすることを意味します。

この手順の一般的な目的は、以下のとおりです。

- 最初に、インストールされているコンテンツを削除します。
- 次に、有効なストリームを変更します。
- 最後に、新しいコンテンツをインストールします。

これにより、依存関係が原因でユーザーが続行できなくなる可能性があるステップが減ります。

前提条件

- モジュールストリームを有効にし、モジュールの別のストリームがある。
- [モジュール依存関係の解決](#) を理解している。
- 元に戻すことができるように、変更した内容を書き留めている。
- パッケージの依存関係が、Red Hat Enterprise Linux の以前のバージョンのものであるため、パッケージおよびモジュールの依存関係によりこの手順が完了できない場合があります。

手順

1. 現在アクティブなモジュールストリームおよびその依存関係からコンテンツを削除します。
 - a. モジュールからインストールしたパッケージの一覧を表示します。

```
$ yum module info module-name | grep module+el8 | sed 's/.*: //g;s/\n/ /g' | xargs yum list installed
```

このモジュールからインストールしたパッケージの一覧を表示します。

- b. 前の手順で一覧表示したパッケージを削除します。

```
# yum remove package
```

package を、前の手順に記載されているパッケージに置き換えます。パッケージが複数になる場合は、その名前をスペースで区切ります。

パッケージの依存関係も、依存パッケージを削除するように強制することがあります。これらのパッケージを書き留め、**yum module provides** コマンドを使用して、どのモジュールがそれを提供しているかを確認し、以降のステップに使用するため、その一覧を保持します。

- c. モジュールプロファイルをアンインストールとしてマークします。

```
# yum module remove module-name
```

2. モジュールおよびそのストリームをリセットします。

```
# yum module reset module-name
```

3. 新しいストリームを有効にし、モジュールの依存関係を解決します。

```
# yum module enable module-name:new-stream
```

yum ツールは、パッケージおよびモジュールの依存関係に必要な概要を提示し、確認を要求します。モジュールの依存関係が、すぐにその操作を行わないようにできます。その場合、この手順を再帰的に適用するか、リセットします。このストリームを見つけるには、このステップのコマンドの出力と、ステップ1で作成した一覧を参照してください。

この手順で続行するには、新しいストリームを有効にする必要があります。組み合わせた依存関係によりそれができない場合は、変更内容を元に戻します。元のストリームを有効にして、元のパッケージをインストールします。

4. モジュールの新しいストリームからコンテンツをインストールします。
 - a. 新しいストリームからプロファイルをインストールします。

```
# yum module install module-name:new-stream/profile
```

- b. 新しいストリームが提供する個々のパッケージをインストールします。

```
# yum module install package
```