



Red Hat Enterprise Linux 8

ユーザー空間コンポーネントの管理および削除

Red Hat Enterprise Linux 8 の AppStream および BaseOS の概要

Red Hat Enterprise Linux 8 ユーザー空間コンポーネントの管理および削除

Red Hat Enterprise Linux 8 の AppStream および BaseOS の概要

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 © 2021 | You need to change the HOLDER entity in the en-US/Installing_managing_and_removing_user-space_components.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、および使用する方法を説明します。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

オープンソースをより包摂的に	3
RED HAT ドキュメントへのフィードバック (英語のみ)	4
第1章 APPSTREAM の使用	5
1.1. RHEL 8 のコンテンツの配布	5
1.2. アプリケーションストリーム	5
1.3. RHEL 8 のパッケージング方法	6
1.4. RHEL 8 で YUM を使用したパッケージ管理	6
第2章 モジュールの概要	7
2.1. モジュールストリーム	7
2.2. モジュールプロファイル	8
第3章 RHEL 8 コンテンツの検索	9
3.1. パッケージの検索	9
3.2. 利用可能なモジュールの一覧表示	9
3.3. コンテンツを一覧表示するコマンド	12
第4章 RHEL 8 コンテンツのインストール	14
4.1. パッケージのインストール	14
4.2. パッケージのインストール前にストリームの選択	15
4.3. モジュールストリームの上書き	15
4.4. モジュールコンテンツのインストール	16
4.5. インストール済みコンテンツの実行	18
4.6. RHEL 8 コンテンツのインストールコマンド	18
4.7. 関連情報	20
第5章 RHEL 8 コンテンツの削除	21
5.1. インストール済みパッケージの削除	21
5.2. インストールされたモジュールコンテンツの削除	21
5.2.1. モジュールストリームから全パッケージを削除	21
5.2.2. インストール済みプロファイルからパッケージを削除	24
5.3. モジュールストリームのリセット	26
5.4. コンテンツを削除するコマンド	27
第6章 APPLICATION STREAM コンテンツのバージョン管理	28
6.1. モジュールの依存関係とストリームの変更	28
6.2. モジュールおよび非モジュールの依存関係のやり取り	29
6.3. モジュールストリームのリセット	29
6.4. モジュールのストリームをすべて無効化	29
6.5. 後続のストリームへの切り替え	29

オープンソースをより包摂的に

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、[弊社](#) の CTO、Chris Wright の [メッセージ](#) を参照してください。

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。改善点を報告する場合は、以下のように行います。

- 特定の文章に簡単なコメントを記入する場合は、以下の手順を行います。
 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端に **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。
- 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 に分類されているかを説明します。
- 「[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られているかを説明します。

1.1. RHEL 8 のコンテンツの配布

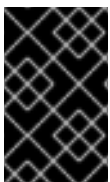
RHEL 8 のコンテンツは、BaseOS と AppStream の 2 つのメインリポジトリにより配布されます。

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。AppStream のコンテンツ形式には、従来の RPM 形式と、RPM 形式の拡張 (**モジュール** と呼ばれます) の 2 つが利用できます。



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 インストールの実行』](#) 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 (ユーザー空間コンポーネントのバージョン) が導入されています。ユーザー空間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 をカスタマイズでき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使用できるコンポーネントは、モジュールまたは RPM パッケージとしてパッケージ化され、Red Hat Enterprise Linux 8 の AppStream リポジトリを介して配信されます。各アプリケーションストリームには、特定のアプリケーションにより適した、RHEL 8 と同じか、より短いライフサイクルが指定されています。ライフサイクルが短いアプリケーションストリームは、「[Red Hat Enterprise Linux 8 Application Streams ライフサイクル](#)」ページに記載されています。



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、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。

関連資料

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

個々の RPM パッケージを扱う YUM の使用方法には変更がありません。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

AppStream リポジトリには、個別の RPM パッケージと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、モジュールのコンテンツを統合して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、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。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、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。

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、サポートされ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

利用可能なモジュールおよびストリームの一覧は『[パッケージマニフェスト](#)』を参照してください。コンポーネント別の変更は、『[リリースノート](#)』を参照してください。

例2.1 postgresql モジュールストリーム

`postgresql` モジュールは、ストリームの 9.6、10、および 12 で、バージョンがそれぞれ 9.6、10、および 12 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。

関連情報

- 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。
- モジュールストリームのアップグレード方法は、「[後続のストリームへの切り替え](#)」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) も含みません。

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。

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3.1「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際に役に立つ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

- 必要な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、プロファイルの一覧、および提供されるパッケージの一覧が含まれます。

```
$ 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。

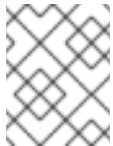
```
$ yum module list module-name
```

関連情報

- 2章 [モジュールの概要](#)

例3.1 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の一覧を表示します。

```
$ yum module list
Name      Stream Profiles  Summary
(...)
postgresql 9.6   client,  PostgreSQL server and client module
          server [d]
postgresql 10 [d] client,  PostgreSQL server and client module
          server [d]
postgresql 12   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
Summary    : PostgreSQL server and client module
Description : (...)
```

```
(...)
Name       : postgresql
Stream    : 12
Version   : 8010120191120141335
Context   : e4e244f9
Profiles  : client, server [d]
Default profiles : server
Repo      : appstream
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
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 module 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 9.6      client, server [d]  (...)
postgresql 10 [d][e] client, server [d] [i] (...)
postgresql 12      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。

3.3. 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。

利用可能なパッケージの一覧表示

```
$ yum list available
```

選択したパッケージで利用可能な YUM リポジトリの検索

```
$ yum repoquery package
```

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、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
```

特定のストリームを使用し、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上書き」](#) では、システムごとにリポジトリモジュールのデフォルトを上書きする方法を説明します。
- [「モジュールコンテンツのインストール」](#) では、モジュール、ストリーム、およびプロファイ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4.3「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。
- [「インストール済みコンテンツの実行」](#) 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は、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。

関連情報

- [「モジュールコンテンツのインストール」](#)
- [「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、そのパッケージ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 により確認が求められ、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であったため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上書き

デフォルトでは、**yum** ユーティリティーは、モジュールを含むリポジトリで定義されているモジュールストリームを使用します。デフォルトのストリームは、`/etc/dnf/modules.defaults.d/` ディレクトリを使って上書きできます。

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

1. `/etc/dnf/modules.defaults.d/` ドロップインディレクトリに YAML 設定ファイルを作成します。

```
---
document: modulemd-defaults
version: 1
data:
  module: postgresql
  stream: "10"
  profiles:
    10: [server]
    12: [server]
    13: [server]
    9.6: [server]
...

```

上記の出力は、この書き込み時の **postgresql** モジュールのデフォルト定義を表しています。

例4.1元のデフォルトを含む **postgresql** モジュールの例

```
# yum module list postgresql
(...)
Red Hat Enterprise Linux 8 for x86_64 - AppStream (RPMs)
Name          Stream    Profiles    Summary
postgresql    9.6      client, server [d]  PostgreSQL server and client module
postgresql    10 [d]   client, server [d]  PostgreSQL server and client module
postgresql    12      client, server [d]  PostgreSQL server and client module
postgresql    13      client, server [d]  PostgreSQL server and client module

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

2. デフォルトのストリームを 13 に設定するには、以下の設定を実装できます。

```
---
document: modulemd-defaults
version: 1
data:
  module: postgresql
  stream: "13"
  profiles:
    10: [server]
    12: [server]
    13: [server]
    9.6: [server]
...

```

例4.2 モジュールのデフォルトが上書きされた postgresql の例

```
# yum module list postgresql
(...)
Red Hat Enterprise Linux 8 for x86_64 - AppStream (RPMs)
Name          Stream    Profiles    Summary
postgresql    9.6      client, server [d]  PostgreSQL server and client module
postgresql    10      client, server [d]  PostgreSQL server and client module
postgresql    12      client, server [d]  PostgreSQL server and client module
postgresql    13 [d]   client, server [d]  PostgreSQL server and client module

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

4.4. モジュールコンテンツのインストール

本セクションでは、モジュールストリームまたはプロファイルが提供するモジュールコンテンツを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

- アクティブなモジュールストリーム (デフォルトのモジュールストリームまたは有効にしたモジュールストリーム) をインストールします。

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

- 選択したモジュールストリームをインストールします。

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

選択したストリームは自動的に有効になります。デフォルトプロファイルがストリームに定義されている場合、このプロファイルは自動的にインストールされます。

- モジュールストリームの選択したプロファイルをインストールします。

```
# yum module 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。

関連情報

- [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](#)」

例4.3 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 (`postgresql-server` パッケージ) 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の一覧を表示して、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 9.6  client, server [d] PostgreSQL server and client module
postgresql 10 [d] client, server [d] PostgreSQL server and client module
postgresql 12  client, server [d] PostgreSQL server and client module
```

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

出力は、ストリームが 9.6、10、および 12 の `postgresql` モジュールが使用できることを示しています。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 module 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、必要なモジュールストリームを **yum** が解決し、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。

デフォルトのストリームを使用するモジュールの有効化

```
# 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。

特定のストリームを使用するモジュールの有効化

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

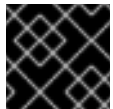
モジュールがデフォルトのストリームを定義する場合は、ストリームとコロンを省略できます。

デフォルトのストリームとプロファイルを使用したモジュールのインストール

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

または、次のようになります。

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



重要

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 module install module-name:stream
```

または、次のようになります。

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

または、次のようになります。

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

4.7. 関連情報

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パッケージのインストール」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 ツールのコマンドの詳細は、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。
- 「[インストールされたモジュールコンテンツの削除](#)」では、モジュールストリームまたはプロファイルから、インストール済み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コマンドをまとめて紹介しません。

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. インストールされたモジュールコンテンツの削除

インストールされたモジュールコンテンツを削除する場合は、選択したプロファイルから、またはストリーム全体からパッケージを削除できます。



重要

`yum` は、依存するパッケージを含む、プロファイルまたはストリームでインストールされたパッケージに対応する名前のパッケージをすべて削除しようとします。(特にシステムでカスタムリポジトリを有効にしている場合は) 続行する前に、削除するパッケージの一覧を常に確認してください。

5.2.1. モジュールストリームから全パッケージを削除

モジュールストリームでインストールされたパッケージを削除すると、他のモジュールで必要なパッケージを除き、ストリームによりインストールされたパッケージに対応する名前のパッケージがすべて削除されます。

前提条件

- モジュールストリームが有効になり、少なくともいくつかのパッケージがストリームからインストールされている。
- [モジュール依存関係の解決](#) を理解している。

手順

1. 選択したストリームからパッケージをすべて削除します。

```
# yum module remove --all module-name:stream
```

module-name および **stream** を、アンインストールするモジュールおよびストリームに置き換えます。

2. 削除トランザクションに進む前に、**Removing:** および **Removing unused dependencies:** にあるパッケージの一覧を確認してください。
3. 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。

選択したプロファイルからのみパッケージを削除する場合は、「インストール済みプロファイルからパッケージを削除」の手順を行ってください。

例5.1 ストリーム全体からパッケージを削除

この例は、モジュールストリームからすべてのパッケージを削除する方法を示しています。

手順

1. すべての利用可能なプロファイルを含む **php:7.3** モジュールストリームをインストールします。

```
[root@rhel-8 ~]# yum module install php:7.3/*
Updating Subscription Management repositories.
Last metadata expiration check: 0:20:19 ago on Tue Mar 3 11:32:05 2020.
Dependencies resolved.
=====
=
Package      Arch Version                               Repository                               Size
=====
=
Installing group/module packages:
libzip      x86_64 1.5.2-1.module+el8.1.0+3189+a1bff096 rhel-8-for-x86_64-
appstream-rpms 63 k
php-cli     x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6 rhel-8-for-x86_64-
appstream-rpms 3.0 M
php-common  x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6 rhel-8-for-x86_64-
appstream-rpms 663 k
php-devel   x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6 rhel-8-for-x86_64-
appstream-rpms 735 k
php-fpm     x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6 rhel-8-for-x86_64-
appstream-rpms 1.6 M
php-json    x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6 rhel-8-for-x86_64-
appstream-rpms 73 k
php-mbstring x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6 rhel-8-for-x86_64-
appstream-rpms 610 k
php-pear    noarch 1:1.10.9-1.module+el8.1.0+3189+a1bff096
rhel-8-for-x86_64-appstream-rpms 359 k
php-pecl-zip x86_64 1.15.4-1.module+el8.1.0+3189+a1bff096
rhel-8-for-x86_64-appstream-rpms 51 k
php-process x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6 rhel-8-for-x86_64-
appstream-rpms 84 k
php-xml     x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6 rhel-8-for-x86_64-
appstream-rpms 188 k
```

```

Installing dependencies:
autoconf      noarch 2.69-27.el8                rhel-8-for-x86_64-appstream-rpms 710
k
...
Installing weak dependencies:
perl-IO-Socket-IP
                noarch 0.39-5.el8                rhel-8-for-x86_64-appstream-rpms 47 k
...
Installing module profiles:
php/common
php/devel
php/minimal
Enabling module streams:
httpd          2.4
nginx          1.14
php            7.3

Transaction Summary
=====
=
Install 73 Packages

Total download size: 76 M
Installed size: 220 M
Is this ok [y/N]: y

```

2. **php:7.3** モジュールストリームからパッケージをすべて削除します。

```

[root@rhel-8 ~]# yum module remove php:7.3 --all
Updating Subscription Management repositories.
Last metadata expiration check: 0:21:26 ago on Tue Mar 3 11:32:05 2020.
Dependencies resolved.
=====
=
Package           Arch  Version                               Repository                               Size
=====
=
Removing:
libzip             x86_64 1.5.2-1.module+el8.1.0+3189+a1bff096
                @rhel-8-for-x86_64-appstream-rpms 313 k
php-cli            x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6
                @rhel-8-for-x86_64-appstream-rpms 11 M
php-common        x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6
                @rhel-8-for-x86_64-appstream-rpms 6.5 M
php-devel         x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6
                @rhel-8-for-x86_64-appstream-rpms 5.3 M
php-fpm           x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6
                @rhel-8-for-x86_64-appstream-rpms 5.6 M
php-json          x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6
                @rhel-8-for-x86_64-appstream-rpms 53 k
php-mbstring      x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6
                @rhel-8-for-x86_64-appstream-rpms 1.9 M
php-pear          noarch 1:1.10.9-1.module+el8.1.0+3189+a1bff096
                @rhel-8-for-x86_64-appstream-rpms 2.1 M
php-pecl-zip      x86_64 1.15.4-1.module+el8.1.0+3189+a1bff096
                @rhel-8-for-x86_64-appstream-rpms 119 k

```

```

php-process      x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6
                  @rhel-8-for-x86_64-appstream-rpms 117 k
php-xml          x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6
                  @rhel-8-for-x86_64-appstream-rpms 557 k
Removing unused dependencies:
autoconf        noarch 2.69-27.el8                @rhel-8-for-x86_64-appstream-rpms 2.2
M
...
Disabling module profiles:
php/common
php/devel
php/minimal

Transaction Summary
=====
=
Remove 73 Packages

Freed space: 220 M
Is this ok [y/N]: y

```

5.2.2. インストール済みプロファイルからパッケージを削除

プロファイルでインストールしたパッケージを削除すると、プロファイルによってインストールされたパッケージに対応する名前を持つパッケージがすべて削除されます。このパッケージの依存関係は、別のプロファイルで必要なパッケージを除きます。

前提条件

- 選択したプロファイルは、**yum module install module-name:stream/profile** コマンドを使用してインストールされているか、**yum install module-name:stream** コマンドを使用してデフォルトのプロファイルとしてインストールされている。
- [モジュール依存関係の解決](#) を理解している。

手順

1. 選択したプロファイルに属するパッケージをアンインストールします。

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

module-name、**stream**、および **profile** を、アンインストールするモジュール、ストリーム、およびプロファイルに置き換えます。

または、ストリームにインストールされているすべてのプロファイルからパッケージをアンインストールします。

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

この操作は、プロファイルに属さないパッケージをストリームから削除しません。

2. 削除トランザクションに進む前に、**Removing:** および **Removing unused dependencies:** にあるパッケージの一覧を確認してください。

選択したストリームからすべてのパッケージを削除するには、「[モジュールストリームから全パッケージを削除](#)」の手順を参照してください。

例5.2 選択したプロファイルからパッケージを削除

この例は、選択したプロファイルにのみ属するパッケージを削除する方法を示しています。

手順

1. すべての利用可能なプロファイルを含む **php:7.3** モジュールストリームをインストールします。

```
[root@rhel-8 ~]# yum module install php:7.3/*
Updating Subscription Management repositories.
Last metadata expiration check: 0:08:41 ago on Tue Mar 3 11:32:05 2020.
Dependencies resolved.
=====
=
Package          Arch  Version                               Repository                               Size
=====
=
Installing group/module packages:
libzip           x86_64 1.5.2-1.module+el8.1.0+3189+a1bff096 rhel-8-for-x86_64-
appstream-rpms 63 k
php-cli          x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6 rhel-8-for-x86_64-
appstream-rpms 3.0 M
php-common       x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6 rhel-8-for-x86_64-
appstream-rpms 663 k
php-devel        x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6 rhel-8-for-x86_64-
appstream-rpms 735 k
php-fpm          x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6 rhel-8-for-x86_64-
appstream-rpms 1.6 M
php-json         x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6 rhel-8-for-x86_64-
appstream-rpms 73 k
php-mbstring     x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6 rhel-8-for-x86_64-
appstream-rpms 610 k
php-pear         noarch 1:1.10.9-1.module+el8.1.0+3189+a1bff096
rhel-8-for-x86_64-appstream-rpms 359 k
php-pecl-zip     x86_64 1.15.4-1.module+el8.1.0+3189+a1bff096
rhel-8-for-x86_64-appstream-rpms 51 k
php-process     x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6 rhel-8-for-x86_64-
appstream-rpms 84 k
php-xml         x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6 rhel-8-for-x86_64-
appstream-rpms 188 k
Installing dependencies:
autoconf        noarch 2.69-27.el8                               rhel-8-for-x86_64-appstream-rpms 710
k
...
Installing weak dependencies:
perl-IO-Socket-IP
noarch 0.39-5.el8                               rhel-8-for-x86_64-appstream-rpms 47 k
...
Installing module profiles:
php/common
php/devel
php/minimal
```

```
Enabling module streams:
```

```
httpd          2.4
nginx          1.14
php            7.3
```

```
Transaction Summary
```

```
=====
```

```
=
Install 73 Packages
```

```
Total download size: 76 M
Installed size: 220 M
Is this ok [y/N]: y
```

2. **devel** プロファイルからパッケージを削除します。

```
[root@rhel-8 ~]# yum module remove php:7.3/devel
Updating Subscription Management repositories.
Last metadata expiration check: 0:09:40 ago on Tue Mar 3 11:32:05 2020.
Dependencies resolved.
```

```
=====
```

```
=
Package          Arch  Version                               Repository          Size
=====
Removing:
libzip            x86_64 1.5.2-1.module+el8.1.0+3189+a1bff096
@rhel-8-for-x86_64-appstream-rpms 313 k
php-devel         x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6
@rhel-8-for-x86_64-appstream-rpms 5.3 M
php-pear          noarch 1:1.10.9-1.module+el8.1.0+3189+a1bff096
@rhel-8-for-x86_64-appstream-rpms 2.1 M
php-pecl-zip      x86_64 1.15.4-1.module+el8.1.0+3189+a1bff096
@rhel-8-for-x86_64-appstream-rpms 119 k
php-process       x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6
@rhel-8-for-x86_64-appstream-rpms 117 k
```

```
Removing unused dependencies:
autoconf         noarch 2.69-27.el8          @rhel-8-for-x86_64-appstream-rpms 2.2
M
...
```

```
Disabling module profiles:
php/devel
```

```
Transaction Summary
```

```
=====
```

```
=
Remove 64 Packages
```

```
Freed space: 193 M
Is this ok [y/N]: y
```

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
```

モジュールは初期状態に戻ります。有効なストリームとインストールされたプロファイルに関する情報は消去されますが、インストールされ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
```

インストール済みプロファイルからパッケージを削除

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

アクティブなストリームからすべてのパッケージを削除

```
# yum module remove --all module-name:stream
```

モジュールを初期状態へのリセット

```
# 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の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。
- 「[後続のストリームへの切り替え](#)」では、モジュールの後続のストリームに変更する手順を説明します。

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、通常は使用する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、その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基本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にしたり、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ために、他のモジュールストリームを無効にする必要がある場合があります。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、「[後続のストリームへの切り替え](#)」の手順に従います。または、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します。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ケージをすべて削除すると、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らないようにします。

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場合は、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の依存関係のやり取り

モジュールの依存関係 は、通常の RPM の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の機能は、リポジトリ間で仮想的な依存関係と同様になります。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。

変更が明示的に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のアップグレードは行いません。

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
```

モジュールは初期状態に戻ります。有効なストリームとインストールされたプロファイルに関する情報は消去されますが、インストールされ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 により確認が求められ、その後、そのすべてのストリームを持つモジュールが無効になります。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. 後続のストリームへの切り替え

後続のモジュールストリームに切り替えると、モジュールのすべてのパッケージが、その後続のバージョンに置き換えられます。



重要

この手順は、以下の条件でのみ使用できます。

前提条件

- システムが完全に更新されている。
- システムにインストールしたパッケージよりも、リポジトリで利用可能なパッケージの方が新しい。

手順

1. 以下のコマンドを実行して、後続のストリームに切り替えるためにシステムの準備が整っているかどうかを判断します。

```
# yum distro-sync
```

このコマンドは、**Nothing to do.Complete!**メッセージで終了する必要があります。代わりに変更を提案し、確認を求められた場合は、この変更を慎重に確認し、続行するかどうかを検討してください。必要に応じて **yum distro-sync** コマンドを繰り返し実行します。または、推奨される変更を受け入れず、システムを、このコマンドが **Nothing to do.Complete!**メッセージを返した状態に手動で戻すことができます。



注記

yum distro-sync の結果を確認してストリームを切り替えると、この手順の最後のステップと同じコマンドが必要になるため、ストリームへの関連外の変更が発生しないようにします。

2. アクティブなストリームを後続のストリームに変更します。

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

3. インストール済みパッケージを同期し、ストリーム間の変更を実行します。

```
# yum distro-sync
```

この動作により、ストリーム外のコンテンツへの変更が提案されている場合は、注意して確認してください。



注記

- 一部のインストール済みパッケージが以前のストリームに依存し、後続のストリームに互換性のあるバージョンがない場合、**yum** は依存関係の競合を報告します。このようなパッケージは、後続のストリームとは依存関係がなく、一緒にインストールすることはできないため、この場合は **--allowerasing** オプションを使用してこのパッケージを削除してください。
- **Perl** モジュールの切り替え時には、RHEL 8 ベースインストールの一部のパッケージが **Perl 5.26** に依存するため、**--allowerasing** オプションが常に必要になります。
- 解釈した言語のバイナリー拡張 (通常は **C** または **C++** で記述) は、新しいストリームを有効にしてから再インストールする必要があります。たとえば、**ruby** モジュールの **gem** コマンド、**nodejs** モジュールの **npm** コマンド、**perl** モジュールの **cpan** コマンド、**php** モジュールの **pecl** コマンドから、特定のパッケージをインストールします。詳細は「[How to switch Ruby streams in RHEL 8](#)」を参照してください。

もしくは、現在のストリームからインストールした **モジュールのコンテンツ** をすべて削除し、**モジュールをリセット** して、**新しいストリームをインストール** します。