



Red Hat Enterprise Linux 8

ユーザー空間コンポーネントのインストール、管理、および削除

YUM ソフトウェア管理ツールを使用した BaseOS リポジトリおよび AppStream リポジトリでのコンテンツの管理

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

YUM ソフトウェア管理ツールを使用した BaseOS リポジトリおよび AppStream リポジトリでのコンテンツの管理

法律上の通知

Copyright © 2024 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

YUM ツールを使用して、BaseOS リポジトリおよび AppStream リポジトリを介して配布されるコンテンツを検索、インストール、および利用します。パッケージ、モジュール、ストリーム、プロファイルの操作方法を学びます。

目次

RED HAT ドキュメントへのフィードバック (英語のみ)	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. APPLICATION STREAMS (APPSTREAM)	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とその内容の一覧表示	8
3.3. コンテンツをリスト表示するコマンド	12
第4章 RHEL 8 コンテンツのインストール	13
4.1. パッケージのインストール	13
4.2. パッケージインストールの前のストリーム選択	14
4.3. モジュールコンテンツのインストール	14
4.4. インストール済みコンテンツの実行	16
4.5. RHEL 8 コンテンツのインストールコマンド	17
4.6. 関連情報	17
第5章 RHEL 8 コンテンツの削除	19
5.1. インストール済みパッケージの削除	19
5.2. インストールしたモジュラーコンテンツの削除	19
5.3. コンテンツを削除するコマンド	26
第6章 APPLICATION STREAM コンテンツのバージョン管理	27
6.1. モジュールの依存関係とストリームの変更	27
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	28
6.3. モジュールストリームのリセット	28
6.4. モジュールのストリームをすべて無効化	28
6.5. 後続のストリームへの切り替え	29
6.6. モジュールのデフォルトストリームの上書き	30

RED HAT ドキュメントへのフィードバック (英語のみ)

Red Hat ドキュメントに関するご意見やご感想をお寄せください。また、改善点があればお知らせください。

Jira からのフィードバック送信 (アカウントが必要)

1. [Jira](#) の Web サイトにログインします。
2. 上部のナビゲーションバーで **Create** をクリックします。
3. **Summary** フィールドにわかりやすいタイトルを入力します。
4. **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追加してください。
5. ダイアログの下部にある **Create** をクリックします。

第1章 APPSTREAM の使用

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

- [RHEL 8 でのコンテンツの配布](#)。
- [アプリケーションストリーム](#)
- [RHEL 8 でのパッケージング方法](#)。
- [RHEL 8 での YUM を使用したパッケージ管理](#)。

1.1. RHEL 8 のコンテンツの配布

Red Hat Enterprise Linux 8 のコンテンツは、**BaseOS** と **AppStream** の 2 つの主要なリポジトリを通じて配布されます。

BaseOS

BaseOS リポジトリには、すべてのインストールの基盤となるオペレーティングシステム機能のコアセットが含まれています。このコンテンツは RPM パッケージの形式で入手でき、Red Hat Enterprise Linux の以前のリリース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をサポートする追加の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 RPM パッケージと、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。インストール手順については、[インストールメディアから RHEL を対話形式でインストールする](#) ドキュメントを参照してください。

1.2. APPLICATION STREAMS (APPSTREAM)

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 リポジトリのコンテンツは、次の2つの方法でパッケージ化されます。

- **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 [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。

次のセクションでは、モジュール内のコンテンツの編成と処理に関する機能を説明します。

- **モジュールストリーム** - バージョンごとのコンテンツの編成。
- **モジュールプロファイル** - 目的別のコンテンツの編成。

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り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。



重要

デフォルトのストリームは、RHEL メジャーリリースでは変更されません。各ストリームの **ライフサイクル** を常に考慮してください。RHEL メジャーリリースの終了前に、デフォルトのストリームがライフサイクルの終了日に到達するインスタンスのデフォルトストリームに依存しないでください。

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リストは [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、12、および 13 で、バージョンがそれぞれ 9.6、10、12、および 13 の PostgreSQL データベースを提供します。ストリーム 10 がデフォルトです。つまり、postgresql を求められた場合、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。

使用するモジュールストリームを常に決定し、バージョンを明示的にインストールします。

関連情報

- [モジュールの依存関係とストリームの変更](#)
- [後のストリームへの切り替え](#)
- [パッケージマニフェスト](#)
- [リリースノート](#)

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。このパッケージリストは、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** - 強化された実稼働対応デプロイメント (デフォルトプロファイル)。
- **devel** - httpd を変更するために必要なパッケージ。
- **最小** - 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

以下のセクションでは、**YUM** を使用して、Red Hat Enterprise Linux 8 の AppStream および BaseOS リポジトリ内のコンテンツを見つけて調べる方法を説明します。

- 目的のコンテンツを提供する [パッケージの検索](#)
- [利用可能なモジュールの一覧表示とその詳細の確認](#)
- [RHEL 8 コンテンツの検索に役立つコマンド](#) を調べます。

3.1. パッケージの検索

特定のアプリケーションまたはその他のコンテンツを提供するパッケージを見つけるには、次の手順を実行します。

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とその内容の一覧表示

使用可能なモジュールとその詳細を確認するには、次の手順を実行します。

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、次を使用します。

```
$ 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。

- 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などモジュールに関する詳細を表示するには、以下を使用します。

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

- 各モジュールプロファイルでインストールされるパッケージの一覧を表示するには、以下を使用します。

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

- 有効なストリームやインストール済みプロファイルなど、モジュールの現在の状態を表示するには、以下を使用します。

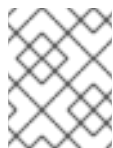
```
$ yum module list module-name
```

関連情報

- [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

以下は、AppStream リポジトリで利用可能なモジュールを一覧表示する方法と、**postgresql** モジュールの内容に関する情報を取得する方法の例です。



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、ここに示すよりも多くの情報が含まれる場合があります。

1. 利用可能なモジュールのリストを表示します。

```
$ 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]
postgresql 13  client, PostgreSQL server and client module
           server [
           d]
postgresql 15  client, PostgreSQL server and client module
           server [
           d]
(...)

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

2. **postgresql** モジュールの詳細を調べます。

```
$ yum module info postgresql

...
Name      : postgresql
Stream    : 10 [d][a]
Version   : 8070020221124143148
Context   : bd1311ed
Architecture : x86_64
Profiles  : client, server [d]
Default profiles : server
Repo      : rhel-AppStream
Summary   : PostgreSQL server and client module
```

```

...

Name       : postgresql
Stream     : 12
Version    : 8060020221003080350
Context    : ad008a3a
Architecture : x86_64
Profiles   : client, server [d]
Default profiles : server
Repo       : rhel-AppStream
Summary    : PostgreSQL server and client module
...

Name       : postgresql
Stream     : 13
Version    : 8070020230227142544
Context    : bd1311ed
Architecture : x86_64
Profiles   : client, server [d]
Default profiles : server
Repo       : rhel-AppStream
Summary    : PostgreSQL server and client module
...

Name       : postgresql
Stream     : 15
Version    : 8080020230212204728
Context    : fd72936b
Architecture : x86_64
Profiles   : client, server [d]
Default profiles : server
Repo       : rhel-AppStream
Summary    : PostgreSQL server and client module
...

Name       : postgresql
Stream     : 9.6
Version    : 8040020210602182503
Context    : 522a0ee4
Architecture : x86_64
Profiles   : client, server [d]
Default profiles : server
Repo       : rhel-AppStream
Summary    : PostgreSQL server and client module
...

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

```

ストリームを指定しない場合には、**yum** は使用可能なすべてのストリームを一覧表示します。

3. **postgresql** モジュールのストリーム **10** で使用可能なプロファイルを調べます。

```

$ yum module info --profile postgresql:10
(...)

```

```
Name : postgresql:10:8070020221124143148:bd1311ed:x86_64
client : postgresql
server : postgresql-server
```

各プロファイルは、依存関係を含む異なるパッケージセットをインストールすることに注意してください。

4. デフォルトのストリーム **10** およびデフォルトプロファイル **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                Architecture Version                Repository            Size
=====
Installing group/module packages:
postgresql-server      x86_64      10.23-1.module+el8.7.0+17280+3a452e1f  rhel-
AppStream              5.1 M
Installing dependencies:
libpq                  x86_64      13.5-1.el8              rhel-AppStream        198 k
postgresql             x86_64      10.23-1.module+el8.7.0+17280+3a452e1f  rhel-
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: 26 M
Is this ok [y/N]: y
...

Installed:
  libpq-13.5-1.el8.x86_64
  postgresql-10.23-1.module+el8.7.0+17280+3a452e1f.x86_64
  postgresql-server-10.23-1.module+el8.7.0+17280+3a452e1f.x86_64

Complete!
```

5. **postgresql** モジュールの現在の状態を確認します。

```
$ yum module list postgresql
rhel-AppStream
Name           Stream      Profiles                Summary
postgresql    9.6         client, server [d]     PostgreSQL server and client
module
postgresql    10 [d][e]   client, server [d] [i] PostgreSQL server and client
module
```

postgresql module	12	client, server [d]	PostgreSQL server and client
postgresql module	13	client, server [d]	PostgreSQL server and client
postgresql module	15	client, server [d]	PostgreSQL server and client

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

出力は、デフォルトストリーム **10** が有効であり、そのプロファイル **サーバー** がインストールされていることを示しています。

3.3. コンテンツをリスト表示するコマンド

以下は、Red Hat Enterprise Linux 8 で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です。

コマンド	説明
yum list available	利用可能なパッケージを一覧表示します。
yum repoquery package	選択したパッケージの利用可能な YUM リポジトリを検索します。
yum search "text string"	任意の文字列でパッケージを検索します。
yum info package	パッケージの詳細を表示します。
yum module provides package	パッケージを提供するモジュールを表示します。 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。
yum module list	利用可能なモジュールを一覧表示します。
yum module info module-name	モジュールの詳細を表示します。
yum module info --profile module-name	デフォルトのストリームを使用して、モジュールのプロファイルによってインストールされたパッケージを一覧表示します。
yum module info --profile module-name:stream	指定されたストリームを使用して、モジュールのプロファイルによってインストールされたパッケージを表示します。
yum module list module-name	モジュールの現在の状態を表示します。

第4章 RHEL 8 コンテンツのインストール

以下のセクションでは、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。

- [パッケージをインストールします。](#)
- [パッケージをインストールするストリームを選択します。](#)
- [モジュール、ストリーム、およびプロファイルによって提供される一連のパッケージをインストールします。](#)
- [RHEL 8 のインストール済みコンテンツを実行します。](#)
- [RHEL 8 コンテンツのインストールに役立つコマンド](#) を調べます。

4.1. パッケージのインストール

パッケージをインストールするには、次の手順を実行します。

手順

- パッケージをインストールします。

```
# yum install package
```

package をパッケージの名前に置き換えます。

- 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としてマークされたモジュールストリームによって提供されている場合には、**yum** はこのパッケージをインストールする前にそのモジュールストリームを自動的に有効にします。



重要

デフォルトのストリームに依存するのではなく、インストール用に特定のモジュールストリームを常に選択することを推奨します。特定のデフォルトのモジュールストリームは、RHEL メジャーリリースが終了する前に、ライフサイクルの終了日に到達します。各ストリームの [ライフサイクル](#) を常に考慮してください。

- 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。

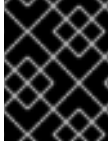
関連情報

- [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](#)

- [Red Hat Enterprise Linux Application Streams Life Cycle](#)

4.2. パッケージインストールの前のストリーム選択

インストールには常に特定のモジュールストリームを選択することを推奨します。各ストリームの [ライフサイクル](#) を常に考慮してください。



重要

特定のデフォルトのモジュールストリームは、RHEL メジャーリリースが終了する前に、ライフサイクルの終了日に到達します。

デフォルト以外のストリームからパッケージをインストールするには、最初にストリームを有効にします。

前提条件

- [アクティブモジュールストリームの概念](#) を理解している。

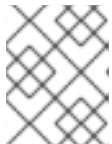
手順

- 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。



注記

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。

関連情報

- [Red Hat Enterprise Linux Application Streams Life Cycle](#)

4.3. モジュールコンテンツのインストール

モジュールストリームまたはプロファイルによって提供されるモジュラーコンテンツをインストールするには、次の手順を実行します。

前提条件

- [アクティブ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

- 選択したモジュールストリームをインストールするには、次を使用します。



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

このコマンドを実行すると、選択したストリームが自動的に有効になります。ストリームに対してデフォルトのプロファイルが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ることに注意してください。



重要

モジュールストリームの [ライフサイクル](#) を常に考慮してください。

- 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するには、以下を使用します。

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

このコマンドを実行すると、ストリームが有効になり、モジュールの特定のストリーム (バージョン) とプロファイル (目的) に対して推奨される一連のパッケージがインストールされます。

関連情報

- [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](#)
- [Red Hat Enterprise Linux Application Streams Life Cycle](#)

例4.1 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

以下は、デフォルト以外のストリーム (バージョン)、つまりバージョン **13** の PostgreSQL サーバー (**postgresql-server** パッケージ) からアプリケーションをインストールする方法の例です。デフォルトのストリームはバージョン **10** を提供します。

手順

- postgresql-server** パッケージを提供するモジュールを一覧表示して、利用可能な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
postgresql 13  client, server [d] PostgreSQL server and client module
postgresql 15  client, server [d] PostgreSQL server and client module
```

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

出力は、**postgresql** モジュールがストリーム **9.6**、**10**、**12**、および **13** で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。

- ストリーム **13** 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:13
...
```

```

Dependencies resolved.
=====
Package                Architecture Version                Repository            Size
=====
Installing group/module packages:
postgresql-server      x86_64      13.10-1.module+el8.7.0+18279+1ca8cf12  rhel-
AppStream              5.6 M
Installing dependencies:
libcicu                x86_64      60.3-2.el8_1          rhel                  8.8 M
libpq                  x86_64      13.5-1.el8            rhel-AppStream       198 k
postgresql             x86_64      13.10-1.module+el8.7.0+18279+1ca8cf12  rhel-
AppStream              1.5 M
Installing module profiles:
postgresql/server
Enabling module streams:
postgresql              13

Transaction Summary
=====
Install 4 Packages

Total download size: 16 M
Installed size: 61 M
Is this ok [y/N]: y

...

Installed:
  libcicu-60.3-2.el8_1.x86_64
  libpq-13.5-1.el8.x86_64
  postgresql-13.10-1.module+el8.7.0+18279+1ca8cf12.x86_64
  postgresql-server-13.10-1.module+el8.7.0+18279+1ca8cf12.x86_64

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) 13.10

```

4.4. 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 8 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。コマンドがモジュールによって有効化された RPM パッケージから作成されたものである場合には、これらのコマンドの使用方法には影響はありません。

手順

- 新しいコマンドを実行するには、コマンドを直接入力します。

```
$ command
```

`command` を、実行するコマンドの名前に置き換えます。

注記

RHEL 8 では、GCC Toolset は Software Collection としてパッケージ化されています。Software Collection としてパッケージ化されたコンポーネントからコマンドを実行するには、次を使用します。

```
$ scl enable collection 'command'
```

`collection` は、ソフトウェアコレクションの名前に置き換えます。

詳細は、[Using GCC Toolset](#) を参照してください。

4.5. RHEL 8 コンテンツのインストールコマンド

以下は、Red Hat Enterprise Linux 8 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コマンドです。

コマンド	説明
<code>yum install package</code>	パッケージのインストール 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せん。
<code>yum module enable module-name:stream</code>	特定のストリームを使用してモジュールを有効にします。 モジュールストリームの ライフサイクル を常に考慮してください。
<code>yum module install module-name:stream</code> <code>yum install @module-name:stream</code>	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
<code>yum module install module-name:stream/profile</code> <code>yum install @module-name:stream/profile</code>	特定のストリームとプロファイルを使用してモジュールをインストールします。

4.6. 関連情報

- [yum を使用したソフトウェアパッケージのインストール](#)
- **yum(8)** man ページ

第5章 RHEL 8 コンテンツの削除

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

- [パッケージの削除](#)
- [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したコンテンツの削除](#)
- [RHEL 8 コンテンツを削除するためのコマンド](#) を調べます。

5.1. インストール済みパッケージの削除

システムにインストールされているパッケージを削除するには、次の手順を実行します。

手順

- 特定のパッケージを削除するには、次を使用します。

```
# yum remove package-name
```

package-name は、削除するパッケージの名前に置き換えます。



注記

yum コマンドは、他の依存パッケージと一緒にパッケージを削除します。

5.2. インストールしたモジュラーコンテンツの削除

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、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を持つすべてのパッケージ (依存パッケージを含む) を削除します。(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リスト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、**yum** は、ストリーム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すべてのパッケージを削除します。これには、他のモジュールが必要とするパッケージを除いて、パッケージの依存関係が含まれます。

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。
- [モジュールの依存関係の解決](#) 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リスト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パッケージのみを削除する場合は、[インストール済みプロファイルからのパッケージの削除](#) の手順に従ってください。

例5.1 ストリーム全体からパッケージを削除

以下は、**php:7.3** 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の例です。

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。

```
# 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 モジュールを検査します。

```

$ yum module info php
...
Name           : php
Stream         : 7.3 [e] [a]
Version        : 8020020200715124551
Context        : ceb1cf90
Architecture   : x86_64
Profiles       : common [d] [i], devel [i], minimal [i]
Default profiles : common
...
Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled, [a]ctive

```

3. 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。

```

# yum module remove --all php:7.3
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

```

4. 削除後に **php** モジュールを検査します。

```

$ yum module info php
...
Name       : php
Stream    : 7.3 [e] [a]
Version   : 8020020200715124551
Context   : ceb1cf90
Architecture : x86_64
Profiles  : common [d], devel, minimal
Default profiles : common
...
Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled, [a]ctive

```

現在、**php** モジュールの **7.3** ストリームが有効になっていますが、このストリームのパッケージはインストールされていません。

5.2.2. インストール済みプロファイルからのパッケージの削除

プロファイルでインストールされたパッケージを削除すると、**yum** はプロファイルでインストールされたパッケージに対応する名前を持つすべてのパッケージを削除します。これには、パッケージの依存関係が含まれます。ただし、別のプロファイルに必要なパッケージは除きます。

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プロファイルとしてインストールされています。
- [モジュールの依存関係の解決](#) 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リスト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の全パッケージの削除](#) の手順に従います。

例5.2 選択したプロファイルからパッケージを削除

以下は、**php:7.3** モジュールストリームの **devel** プロファイルに属するパッケージとその依存関係を削除する方法の例です。



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、ここに示すよりも多くの情報が含まれる場合があります。

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。

```
# 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. インストールされたプロファイルを検査します。

```
$ yum module info php
```

```

...
Name       : php
Stream     : 7.3 [e] [a]
Version    : 8020020200715124551
Context    : ceb1cf90
Architecture : x86_64
Profiles   : common [d] [i], devel [i], minimal [i]
Default profiles : common
Repo       : rhel-AppStream
...
Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled, [a]ctive

```

出力に示されているように、すべてのプロファイルがインストールされます。

3. **devel** プロファイルからパッケージを削除します。

```

# 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

```

4. 削除後にインストールされたプロファイルを検査します。

```

$ yum module info php
...
Name       : php

```

```

Stream      : 7.3 [e] [a]
Version     : 8020020200715124551
Context     : ceb1cf90
Architecture : x86_64
Profiles    : common [d] [i], devel, minimal [i]
Default profiles : common
Repo        : rhel-AppStream
...
Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled, [a]ctive

```

devel を除くすべてのプロファイルが現在インストールされています。

5.3. コンテンツを削除するコマンド

以下は、Red Hat Enterprise Linux 8 でコンテンツを削除するためによく使用されるコマンドです。

コマンド	説明
yum remove package	パッケージを削除します。
yum module remove module-name:stream/profile	インストール済みプロファイルからのパッケージを削除します。
yum モジュールの削除 --all モジュール名: ストリーム	Remove all packages from an active stream.
yum module reset module-name	モジュールを初期状態へのリセット
yum module disable module-name	モジュールおよびそのストリームの無効化

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。

以下のセクションでは、既存の有効なモジュールストリームを変更するときに行う必要がある操作について説明します。

- [モジュールの依存関係とストリームの変更。](#)
- [モジュールおよび非モジュールの依存関係のやり取り。](#)
- [モジュールを初期状態へのリセット。](#)
- [モジュールとそのすべてのストリームの完全な無効化](#)
- [モジュールの後のストリームへの切り替え。](#)
- [モジュールのデフォルトストリームのオーバーライド。](#)

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ことが必要になる場合があります。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、今後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 リポジトリの動作を反映しています。



重要

同じモジュールの別のストリームがすでに有効になっている場合、モジュールのストリームを有効にすることはできません。ストリームを切り替えるには、[後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。

システムは、明示的に変更するように指示されない限り、モジュールとストリームの選択を常に保持します。モジュラーパッケージは、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るバージョンに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、モジュールをリセットすると、こ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
```

module-name は、リセットするモジュールの名前に置き換えます。

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を持つモジュールでは、常に1つのストリームがアクティブになります。すべてのモジュールストリームのコンテンツにアクセスしてはならない場合は、モジュール全体を無効にすることができます。

前提条件

- [アクティブ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。

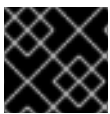
```
# yum module disable module-name
```

module-name は、無効にするモジュールの名前に置き換えます。

yum コマンドにより確認が求められた後に、モジュールとそのすべてのストリームが無効になります。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、前提条件セクションで説明されている条件下でのみ実行できます。

前提条件

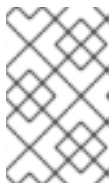
- システムが完全に更新されている。
- システムにインストールしたパッケージよりも、リポジトリで利用可能なパッケージの方が新しい。

手順

1. システムが後のストリームに切り替える準備ができているかどうかを確認します。

```
# yum distro-sync
```

このコマンドは、**Nothing to do.Complete!**の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。



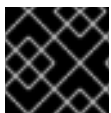
注記

- 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** オプションを常に使用する必要があります。
- 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。

6.6. モジュールのデフォルト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たモジュールのデフォルトストリームを使用します。`/etc/dnf/modules.defaults.d/` ディレクトリでデフォルトストリームをオーバーライドできます。



重要

モジュールストリームの [ライフサイクル](#) を常に考慮してください。

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]
15: [server]
9.6: [server]

```

...

上記の出力は、この記事の執筆時点で **postgresql** モジュールに存在するデフォルトの定義を表しています。

例6.1元のデフォルトを含む postgresql モジュールの例

以下は、**postgresql** モジュールのストリーム **13** をデフォルトストリームとして設定する方法の例です。

1. **postgresql** モジュールを調べます。

```

# 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
postgresql    15     client, server [d]  PostgreSQL server and client module
...
Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled

```

2. デフォルトストリームを **13** に設定するには、次の YAML ファイル設定を **/etc/dnf/modules.defaults.d/postgresql.yaml** ファイルに実装します。

```

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

```

3. **postgresql** モジュールをもう一度調べます。

```

# 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

```

```
postgresql    15      client, server [d]  PostgreSQL server and client module
...
Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled
```