Red Hat Enterprise Linux 8

Installing, managing, and removing user-space components

Managing content in the BaseOS and AppStream repositories by using the YUM software management tool

Last Updated: 2024-03-20

Red Hat Enterprise Linux 8 Installing, managing, and removing user-space components

Managing content in the BaseOS and AppStream repositories by using the YUM software management tool

Legal Notice

Abstract

Find, install, and utilize content distributed through the BaseOS and AppStream repositories by using the YUM tool. Learn how to work with packages, modules, streams, and profiles.

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright's message.

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your feedback on our documentation. Let us know how we can improve it.

Submitting feedback through Jira (account required)

1. Log in to the Jira website.
2. Click **Create** in the top navigation bar.
3. Enter a descriptive title in the **Summary** field.
4. Enter your suggestion for improvement in the **Description** field. Include links to the relevant parts of the documentation.
5. Click **Create** at the bottom of the dialogue.

CHAPTER 1. USING APPSTREAM

In the following sections, learn the concepts related to the AppStream repository in Red Hat Enterprise Linux 8:

- Distribution of content in RHEL 8 .
- Application Streams.
- Packaging methods in RHEL 8 .
- Package management using YUM in RHEL 8 .

1.1. DISTRIBUTION OF CONTENT IN RHEL 8

Red Hat Enterprise Linux 8 content is distributed through the two main repositories: **BaseOS** and **AppStream**.

BaseOS

The BaseOS repository contains the core set of the underlying operating system functionality that provides the foundation for all installations. This content is available in the form of RPM packages and is subject to support terms similar to those in earlier releases of Red Hat Enterprise Linux.

AppStream

The AppStream repository contains additional user-space applications, runtime languages, and databases in support of the varied workloads and use cases. Content in AppStream is available in one of two formats - the RPM packages and an extension to the RPM format called *modules*.

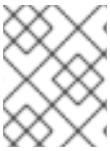
IMPORTANT

Both BaseOS and AppStream content sets are required for a basic RHEL installation, and are available with all RHEL subscriptions. For installation instructions, see the Performing a standard RHEL 8 installation document.

1.2. APPLICATION STREAMS

Red Hat Enterprise Linux 8 introduces the concept of Application Streams - versions of user-space components. Multiple versions of these components are now delivered and updated more frequently than the core operating system packages. This provides greater flexibility to customize Red Hat Enterprise Linux without impacting the underlying stability of the platform or specific deployments.

Components made available as Application Streams can be packaged as modules or RPM packages, and are delivered through the AppStream repository in Red Hat Enterprise Linux 8. Each Application Stream has a given life cycle, either the same as RHEL 8 or shorter, more suitable to the particular application. Application Streams with a shorter life cycle are listed in the Red Hat Enterprise Linux 8 Application Streams Life Cycle page.



NOTE

Not all modules are Application Streams. Dependencies of other modules are not considered Application Streams.

Additional Resources

- Red Hat Enterprise Linux Life Cycle
- Red Hat Enterprise Linux 8 Application Streams Life Cycle

1.3. PACKAGING METHODS IN RHEL 8

Content in the AppStream repository is packaged in two ways:

- **Individual RPM packages**
  Traditional RPM packages available for immediate installation.
- **Modules**
  Modules are collections of packages representing a logical unit: an application, a language stack, a database, or a set of tools. These packages are built, tested, and released together.

1.4. PACKAGE MANAGEMENT USING YUM IN RHEL 8

The **YUM** package management tool is now based on the DNF technology and it adds support for the new modular features.

Usage of **YUM** has not been changed when handling individual RPM packages. For handling the modular content, the **yum module** command has been added. See Installing RHEL 8 content for additional details.

Where required, the modular functionality automatically selects the appropriate combination of modules and streams to enable installation of logical sets of packages for convenient usage.

CHAPTER 2. INTRODUCTION TO MODULES

Besides individual RPM packages, the AppStream repository contains modules. A module is a set of RPM packages that represent a component and are usually installed together. A typical module contains packages with an application, packages with the application-specific dependency libraries, packages with documentation for the application, and packages with helper utilities.

In the following sections, learn features for organization and handling of content within modules:

- Module streams - organization of content by version.
- Module profiles - organization of content by purpose.

2.1. MODULE STREAMS

Module streams are filters that can be imagined as virtual repositories in the AppStream physical repository. Module streams represent versions of the AppStream components. Each of the streams receives updates independently.

Module streams can be active or inactive. Active streams give the system access to the RPM packages within the particular module stream, allowing installation of the respective component version. Streams are active either if marked as default or if they are explicitly enabled by a user action.

Only one stream of a particular module can be active at a given point in time. Therefore, only one version of a component can be installed on a system. Different versions can be used in separate containers.

Each module can have a default stream. Default streams make it easy to consume RHEL packages the usual way without the need to learn about modules. The default stream is active, unless the whole module has been disabled or another stream of that module enabled.

IMPORTANT

The default stream does not change throughout the RHEL major release. Always consider each stream's life cycle. Do not rely on the default stream for instances in which the default stream reaches the End of Life status prior to the end of the RHEL major release.

Certain module streams depend on other module streams. For example, the **perl-App-cpanminus**, **perl-DBD-MySQL**, **perl-DBD-Pg**, **perl-DBD-SQLite**, **perl-DBI**, **perl-YAML**, and **freeradius** module streams depend on certain **perl** module streams.

To select a particular stream for a runtime user application or a developer application, consider the following:

- Required functionality and which component versions support it
- Compatibility
- Life cycle length and your update plan

For a list of all available modules and streams, see the Package manifest. For per-component changes, see the Release Notes.

Example 2.1. postgresql module streams

> The **postgresql** module provides the **PostgreSQL** database versions 9.6, 10, 12, and 13 in the respective streams **9.6**, **10**, **12**, **13**, and **15**. Stream **10** is the default one. This means that the system attempts to install the **postgresql-10.6** package if asked for **postgresql**.

Always decide which module stream you want to use, and install the version explicitly.

Additional resources

- Modular dependencies and stream changes
- Switching to a later stream
- Package manifest
- Release Notes

2.2. MODULE PROFILES

A profile is a list of recommended packages that are installed together for a particular use case, such as a server, client, development, minimal install, or other. These package lists can contain packages outside the module stream, usually from the BaseOS repository or the dependencies of the stream.

Installing packages by using a profile is a one-time action provided for the user's convenience. It does not prevent installing or uninstalling any of the packages provided by the module. It is also possible to install packages by using multiple profiles of the same module stream without any further preparatory steps.

Each module stream can have any number of profiles, including none. For any given module stream, some of its profiles can be marked as *default* and are then used for profile installation actions when no profile is explicitly specified. However, existence of a default profile for a module stream is not required.

> **Example 2.2. httpd module profiles**
>
> The **httpd** module, which provides the **Apache** web server, offers the following profiles for installation:
>
> - **common** - a hardened production-ready deployment, the default profile.
> - **devel** - the packages necessary for making modifications to **httpd**.
> - **minimal** - the smallest set of packages that provide a running web server.

CHAPTER 3. FINDING RHEL 8 CONTENT

In the following sections, learn how to locate and examine content in the AppStream and BaseOS repositories in Red Hat Enterprise Linux 8 by using **YUM**:

- Search for packages providing desired content.
- List available modules and find out details about them .
- Examine useful commands for inspecting RHEL 8 content .

3.1. SEARCHING FOR A PACKAGE

To find a package providing a particular application or other content, complete the following steps.

Procedure

1. Search for a package with a text string, such as application name:

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

2. View details about a package:

```
$ yum info package
```

3.2. LISTING AVAILABLE MODULES AND THEIR CONTENT

To find out which modules are available and what their details are, complete the following steps.

Procedure

- To list module streams available to your system, use:

```
$ yum module list
```

  The output of this command lists module streams with name, stream, profiles, and summary on a separate line.

- To display details about a module, including a description, a list of all profiles, and a list of all provided packages, use:

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

- To list which of these packages are installed by each of module profiles, use:

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

- To display the current status of a module, including enabled streams and installed profiles, use:

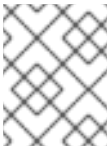
```
$ yum module list module-name
```

Additional resources

- Introduction to modules

Example 3.1. Finding out details about a module

The following is an example of how to list available modules in the AppStream repository and how to obtain information about the **postgresql** module's contents.



NOTE

The outputs in this example have been edited for brevity. Actual outputs might contain more information than shown here.

1. List available modules:

```
$ 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. Examine details of the **postgresql** module:

```
$ 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
```

If you do not specify any stream, **yum** lists all available streams.

3. Examine profiles available in stream **10** of the **postgresql** module:

```
$ yum module info --profile postgresql:10
(...)
Name   : postgresql:10:8070020221124143148:bd1311ed:x86_64
client : postgresql
server : postgresql-server
```

Note that each of the profiles installs a different set of packages, including their dependencies.

4. Install the **postgresql** module by using the default stream **10** and the default profile **server**:

```
# 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. Inspect the current status of the **postgresql** module:

```
$ 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          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
```

The output shows that the default stream **10** is enabled and its profile **server** is installed.

3.3. COMMANDS FOR LISTING CONTENT

The following are the commonly used commands for finding content and its details in Red Hat Enterprise Linux 8.

Command	Description
yum list available	List available packages.
yum repoquery *package*	Search available **YUM** repositories for a selected package.
yum search *"text string"*	Search for a package by using arbitrary text string.
yum info *package*	Display details for a package.
yum module provides *package*	Display which modules provide a package. If the package is available outside any modules, the output of this command is empty.
yum module list	List available modules.
yum module info *module-name*	Display details of a module.
yum module info --profile *module-name*	List packages installed by profiles of a module by using the default stream.
yum module info --profile *module-name*:*stream*	Display packages installed by profiles of a module by using a specified stream.
yum module list *module-name*	Display the current status of a module.

CHAPTER 4. INSTALLING RHEL 8 CONTENT

In the following sections, learn how to install content in Red Hat Enterprise Linux 8:

- Install a package.
- Select a stream for package installation.
- Install sets of packages provided by modules, streams, and profiles .
- Run RHEL 8 installed content .
- Examine useful commands for installing RHEL 8 content .

4.1. INSTALLING A PACKAGE

To install a package, complete the following steps.

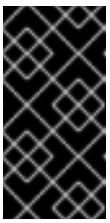
Procedure

- Install a package:

```
# yum install package
```

  Replace *package* with the name of the package.

  - If the package is not provided by any module stream, this procedure is identical to the procedure used on earlier versions of Red Hat Enterprise Linux.
  - If the package is provided by an module stream that is enabled, the package is installed without any further manipulation.
  - If the package is provided by a module stream marked as default, **yum** automatically enables that module stream before installing this package.



IMPORTANT

It is recommended to always select a specific module stream for installation instead of relying on the default stream. Certain default module streams reach the End of Life status prior to the end of the RHEL major release. Always consider each stream's lifecycle.

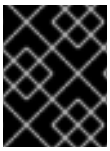
- If the package is provided by a module stream that is not active (neither of the above cases), it is not recognized until you manually enable the respective module stream.

Additional resources

- Installing modular content
- Package management using YUM in RHEL 8
- Red Hat Enterprise Linux Application Streams Life Cycle

4.2. SELECTING A STREAM BEFORE INSTALLATION OF PACKAGES

It is recommended to always select a specific module stream for installation. Always consider each stream's life cycle.



IMPORTANT

Certain default module streams reach the End of Life status prior to the end of the RHEL major release.

To install packages from a non-default stream, enable the stream first.

Prerequisites

- You understand the concept of an *active module stream*.

Procedure

- Enable the module stream:

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

Replace *module-name* and *stream* with names of the module and stream.

yum asks for confirmation and the stream is enabled and active.

NOTE

If another stream of the module was previously active because it was default, it is no longer active.

Additional resources

- Red Hat Enterprise Linux Application Streams Life Cycle

4.3. INSTALLING MODULAR CONTENT

To install modular content provided by a module stream or a profile, complete the following steps.

Prerequisites

- You understand the concept of an *active module stream*.
- You do not have any packages installed from another stream of the same module.

Procedure

- To install a selected module stream, use:

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

By running this command, you automatically enable selected stream. Note that if a default profile is defined for the stream, this profile is automatically installed.

IMPORTANT

Always consider the module stream's life cycle.

- To install a selected profile of the module stream, use:

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

By running this command, you enable the stream and install the recommended set of packages for a given stream (version) and profile (purpose) of the module.

Additional resources

- Introduction to modules
- Commands for installing RHEL 8 content
- Red Hat Enterprise Linux Application Streams Life Cycle

Example 4.1. Installing a non-default stream of an application

The following is an example of how to install an application from a non-default stream (version), namely, the **PostgreSQL** server (the **postgresql-server** package) in version **13**. The default stream provides version **10**.

Procedure

1. List modules that provide the **postgresql-server** package to see which streams are available:

```
$ 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
```

The output shows that the **postgresql** module is available with streams **9.6**, **10**, **12**, **13**, and **15**. The default stream is **10**.

2. Install the packages provided by the **postgresql** module in stream **13**:

```
# 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:
 libicu                  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:
  libicu-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!
```

Because the installation profile was not specified, the default profile **server** was used.

3. Verify the installed version of **PostgreSQL**:

```
$ postgres --version
postgres (PostgreSQL) 13.10
```

4.4. RUNNING INSTALLED CONTENT

New commands are usually enabled after you install content from RHEL 8 repositories. If the commands originated from RPM packages that were enabled by a module, the experience of using these command should be no different.

Procedure

- To run the new commands, enter them directly:

```
$ command
```

Replace *command* with the name of the command you want to run.

NOTE

In RHEL 8, GCC Toolset is packaged as a Software Collection. To run a command from a component packaged as a Software Collection, use:

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

Replace *collection* with the name of the Software Collection.

For more information, see Using GCC Toolset.

4.5. COMMANDS FOR INSTALLING RHEL 8 CONTENT

The following are the commonly used commands for installing Red Hat Enterprise Linux 8 content.

Command	Description
yum install *package*	Install a package. If the package is provided by a module stream, **yum** resolves the required module stream and enables it automatically while installing this package. This also happens recursively for all package dependencies. If more module streams satisfy the requirement, the default ones are used.
yum module enable *module-name*:*stream*	Enable a module by using a specific stream. Always consider the module stream's life cycle.
yum module install *module-name*:*stream* **yum install @*module-name*:*stream***	Install a module by using a specific stream and default profiles
yum module install *module-name*:*stream*/*profile* **yum install @*module-name*:*stream*/*profile***	Install a module by using a specific stream and profile.

4.6. ADDITIONAL RESOURCES

- Installing software packages with yum
- **yum(8)** man page

CHAPTER 5. REMOVING RHEL 8 CONTENT

In the following sections, learn how to remove content in Red Hat Enterprise Linux 8:

- Remove a package.
- Remove content installed from a module stream or a profile .
- Examine commands for removing RHEL 8 content .

5.1. REMOVING INSTALLED PACKAGES

To remove a package installed on your system, complete the following steps.

Procedure

- To remove a specific package, use:

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

Replace *package-name* with the name of the package you want to remove.

NOTE

The **yum** command removes a package together with any other dependent packages.

5.2. REMOVING INSTALLED MODULAR CONTENT

When removing installed modular content, you can remove packages either from a selected profile or from the whole stream.

IMPORTANT

YUM removes all packages with a name corresponding to the packages installed with a profile or a stream, including their dependent packages. Always check the list of packages to be removed before you proceed, especially if you have enabled custom repositories on your system.

5.2.1. Removing all packages from a module stream

When you remove packages installed with a module stream, **yum** removes all packages with a name corresponding to the packages installed by the stream. This includes packages' dependencies, with the exception of packages required by other modules.

Prerequisites

- The module stream has been enabled and at least some packages from the stream have been installed.
- You understand modular dependency resolution.

Procedure

1. Remove all packages from a selected stream:

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

   Replace *module-name* and *stream* with the module and stream you want to uninstall.

2. Check the list of packages under **Removing:** and **Removing unused dependencies:** before you proceed with the removal transaction.

3. Optionally, reset or disable the stream.

If you want to remove only packages from a selected profile, follow instructions in Removing packages from an installed profile.

Example 5.1. Removing packages from the whole stream

The following is an example of how to remove all packages from the **php:7.3** module stream.

Procedure

1. Install the **php:7.3** module stream, including all available profiles:

```
# 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. Inspect the **php** module:

```
$ 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. Remove all packages from the **php:7.3** module stream:

```
# 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. Inspect the **php** module after the removal:

```
$ 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
```

The **7.3** stream of the **php** module is currently enabled but no packages from this streams are installed.

5.2.2. Removing packages from an installed profile

When you remove packages installed with a profile, **yum** removes all packages with a name corresponding to the packages installed by the profile. This includes package dependencies, with the exception of packages required by a different profile.

Prerequisites

- The selected profile has been installed by using the **yum module install *module-name*:*stream*/*profile*** command or as a default profile by using the **yum install *module-name*:*stream*** command.
- You understand modular dependency resolution.

Procedure

1. Uninstall packages belonging to the selected profile:

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

   Replace *module-name*, *stream*, and *profile* with the module, stream, and profile you want to uninstall.

   Alternatively, uninstall packages from all installed profiles within a stream:

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

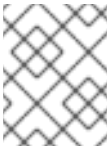
   These operations will not remove packages from the stream that do not belong to any of the profiles.

2. Check the list of packages under **Removing:** and **Removing unused dependencies:** before you proceed with the removal transaction.

To remove all packages from a selected stream, follow instructions in Removing all packages from a module stream.

Example 5.2. Removing packages from a selected profile

The following is an example of how to remove packages and their dependencies that belong to the **devel** profile of the **php:7.3** module stream.



NOTE

The outputs in this example have been edited for brevity. Actual outputs might contain more information than shown here.

Procedure

1. Install the **php:7.3** module stream, including all available profiles:

```
# 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. Inspect the installed profiles:

```
$ 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
```

All profiles are installed as indicated in the output.

3. Remove packages from the **devel** profile:

```
# 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. Inspect the installed profiles after the removal:

```
$ 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
```

All profiles except **devel** are currently installed.

5.3. COMMANDS FOR REMOVING CONTENT

The following are the commonly used commands for removing content in Red Hat Enterprise Linux 8.

Command	Description
yum remove *package*	Remove a package.
yum module remove *module-name*:*stream*/*profile*	Remove packages from an installed profile.
yum module remove --all *module-name*:*stream*	Remove all packages from an active stream.
yum module reset *module-name*	Reset a module to the initial state.
yum module disable *module-name*	Disable a module and all its streams.

CHAPTER 6. MANAGING VERSIONS OF APPLICATION STREAM CONTENT

Content in the AppStream repository can be available in multiple versions, corresponding to module streams.

In the following sections, learn operations you must perform when changing existing enabled module streams:

- Modular dependencies and stream changes.
- Interaction of modular and non-modular dependencies.
- Resetting modules to their initial state .
- Completely disable a module and all its streams .
- Switch to a later stream of a module.
- Override module default streams.

6.1. MODULAR DEPENDENCIES AND STREAM CHANGES

Traditionally, packages providing content depend on further packages, and usually specify the desired dependency versions. For packages contained in modules, this mechanism applies as well, but the grouping of packages and their particular versions into modules and streams provides further constraints. Additionally, module streams can declare dependencies on streams of other modules, independent of the packages contained and provided by them.

After any operations with packages or modules, the whole dependency tree of all underlying installed packages must satisfy all the conditions the packages declare. Additionally, all module stream dependencies must be satisfied.

As a result:

- Enabling a module stream can require enabling streams of further modules.
- Installing a module stream profile or installing packages from a stream can require enabling streams of further modules and installing further packages.
- Disabling a stream of a module can require disabling other module streams. No packages will be removed automatically.
- Removing a package can require removing further packages. If these packages were provided by modules, the module streams remain enabled in preparation for further installation, even if there are no packages installed from these streams anymore. This mirrors the behavior of an unused **YUM** repository.

IMPORTANT

You cannot enable a stream of a module when another stream of the same module is already enabled. To switch streams, follow the procedure in Switching to a later stream . Alternatively, reset the module, and then enable the new stream.

Removing all packages installed from a stream before switching to a different stream prevents the system from reaching states where packages could be installed with no repository or stream that provides them.

Technically, resetting module does not automatically change any installed packages. Removing the packages provided by the previous stream and any packages that depend on them is an explicit manual operation.

6.2. INTERACTION OF MODULAR AND NON-MODULAR DEPENDENCIES

Modular dependencies are an additional layer on top of regular RPM dependencies. Modular dependencies behave similarly to hypothetical dependencies between repositories. This means that installing different packages requires not only resolution of the RPM dependencies, but also the modular dependencies must be resolved beforehand.

The system always retains the module and stream choices, unless explicitly instructed to change them. A modular package receives updates contained in the currently enabled stream of the module that provides this package, but does not upgrade to a version contained in a different stream.

6.3. RESETTING MODULE STREAMS

Resetting a module is an action that returns all of its streams to their initial state - neither enabled nor disabled. If the module has a default stream, this stream becomes active as a result of resetting the module.

Procedure

- Reset the module state:

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

  Replace *module-name* with the name of the module that you want to reset.

  The module is returned to the initial state. Information about an enabled stream and installed profiles is erased but no installed content is removed.

6.4. DISABLING ALL STREAMS OF A MODULE

Modules that have a default stream always have one stream active. In situations where the content from all the module streams must not be accessible, it is possible to disable the whole module.

Prerequisites

- You understand the concept of an *active module stream*.

Procedure

- Disable the module:

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

Replace *module-name* with the name of the module that you want to disable.

The **yum** command asks for confirmation and then disables the module with all its streams. All of the module streams become inactive. No installed content is removed.

6.5. SWITCHING TO A LATER STREAM

When you switch to a later module stream, all packages from the module are replaced with their later versions.

IMPORTANT

This procedure is feasible only under the conditions described in the Prerequisites section.

Prerequisites

- The system is fully updated.
- No packages installed on the system are newer than the packages available in the repository.

Procedure

1. Determine if your system is prepared for switching to a later stream:

```
# yum distro-sync
```

This command must finish with the message *Nothing to do. Complete!* If it instead proposes changes and asks for confirmation, carefully review these changes and consider whether you want to proceed. Run the **yum distro-sync** command repeatedly if necessary. Alternatively, you can refuse the suggested changes and manually modify your system to a state where the command returns *Nothing to do. Complete!*

NOTE

By checking the **yum distro-sync** result before switching the streams, you prevent making changes to the system that are unrelated to the stream switching because the same command is required as the last step of this procedure.

2. Change the active stream to the later one:

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

3. Synchronize installed packages to perform the change between streams:

```
# yum distro-sync
```

If this action suggests changes to content outside the streams, review them carefully.

NOTE

- If certain installed packages depend on the earlier stream, and there is no compatible version in the later stream, **yum** reports a dependency conflict. In this case, use the **--allowerasing** option to remove such packages because they cannot be installed together with the later stream due to missing dependencies.
- When switching **Perl** modules, you must always use the **--allowerasing** option because certain packages in the base RHEL 8 installation depend on **Perl 5.26**.
- Binary extensions (typically written in C or C++) for interpreted languages need to be reinstalled after the new stream is enabled; for example, certain packages installed by the **gem** command from the **ruby** module, the **npm** command from the **nodejs** module, the **cpan** command from the **perl** module, or the **pecl** command from the **php** module. For more information, see How to switch Ruby streams in RHEL 8 .

Alternatively, remove all the module's content installed from the current stream, reset the module, and install the new stream.

6.6. OVERRIDING MODULE DEFAULT STREAMS

By default, the **YUM** utility uses the module default streams defined in the repository that contains the modules. You can override the default stream in the **/etc/dnf/modules.defaults.d/** directory.

IMPORTANT

Always consider the module stream's life cycle.

Prerequisites

- You understand the concept of an active module stream .

Procedure

1. Create a YAML configuration file in the **/etc/dnf/modules.defaults.d/** drop-in directory.

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

The preceding output represents the default definition present for the **postgresql** module at the time of this writing.

Example 6.1. Example postgresql module with original defaults

The following is an example of how to configure the stream **13** of the **postgresql** module as the default stream.

1. Examine the **postgresql** module:

```
# 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. To set the default stream to **13**, implement the following YAML file configuration in the **/etc/dnf/modules.defaults.d/postgresql.yaml** file.

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

3. Examine the **postgresql** module again:

```
# 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
```