



Red Hat Enterprise Linux 9

Gérer les logiciels avec l'outil DNF

Gestion du contenu des dépôts RPM à l'aide de l'outil de gestion de logiciels DNF

Red Hat Enterprise Linux 9 Gérer les logiciels avec l'outil DNF

Gestion du contenu des dépôts RPM à l'aide de l'outil de gestion de logiciels DNF

Notice légale

Copyright © 2023 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.

Résumé

Trouver, installer et utiliser le contenu distribué par les dépôts RPM à l'aide de l'outil DNF.

Apprendre à travailler avec des paquets, des modules, des flux et des profils.

Table des matières

RENDRE L'OPEN SOURCE PLUS INCLUSIF	4
FOURNIR UN RETOUR D'INFORMATION SUR LA DOCUMENTATION DE RED HAT	5
CHAPITRE 1. OUTILS DE GESTION DES LOGICIELS DANS RED HAT ENTERPRISE LINUX 9	6
CHAPITRE 2. DISTRIBUTION DU CONTENU DANS RHEL 9	7
2.1. RÉFÉRENTIELS	7
2.2. FLUX D'APPLICATIONS	7
2.3. MODULES	8
2.4. FLUX DE MODULES	8
2.5. PROFILS DES MODULES	9
CHAPITRE 3. CONFIGURATION DE LA DNF	11
3.1. VISUALISATION DES CONFIGURATIONS DNF ACTUELLES	11
3.2. RÉGLAGE DES OPTIONS PRINCIPALES DE DNF	11
3.3. UTILISATION DES PLUG-INS DNF	11
CHAPITRE 4. RECHERCHE DE CONTENU SUR RHEL 9	13
4.1. RECHERCHE DE PROGICIELS	13
4.2. LISTER LES PROGICIELS	14
4.3. LISTER LES RÉFÉRENTIELS	14
4.4. AFFICHAGE DES INFORMATIONS SUR LE PAQUET	15
4.5. LISTER LES GROUPES DE PAQUETS	15
4.6. LISTE DES MODULES DISPONIBLES ET DE LEUR CONTENU	16
4.7. SPÉCIFICATION D'EXPRESSIONS GLOBALES DANS L'ENTRÉE DNF	19
4.8. RESSOURCES SUPPLÉMENTAIRES	19
CHAPITRE 5. INSTALLATION DU CONTENU DE RHEL 9	20
5.1. INSTALLATION DES PAQUETS	20
5.2. INSTALLATION DE GROUPES DE PAQUETS	21
5.3. INSTALLER UN CONTENU MODULAIRE	21
5.4. EXÉCUTION DU CONTENU INSTALLÉ	23
5.5. CONFIGURATION DE FLUX ET DE PROFILS DE MODULES PERSONNALISÉS PAR DÉFAUT	24
5.6. RESSOURCES SUPPLÉMENTAIRES	25
CHAPITRE 6. MISE À JOUR DU CONTENU DE RHEL 9	26
6.1. VÉRIFICATION DES MISES À JOUR	26
6.2. MISE À JOUR DES PAQUETS	26
6.3. MISE À JOUR DES PAQUETS LIÉS À LA SÉCURITÉ	27
CHAPITRE 7. AUTOMATISATION DES MISES À JOUR LOGICIELLES DANS RHEL 9	28
7.1. INSTALLATION DE DNF AUTOMATIC	28
7.2. DNF FICHER DE CONFIGURATION AUTOMATIQUE	28
7.3. ACTIVATION DU DNF AUTOMATIQUE	29
7.4. VUE D'ENSEMBLE DES UNITÉS TIMER DE SYSTEMD INCLUSES DANS LE PAQUETAGE DNF-AUTOMATIC	31
CHAPITRE 8. SUPPRESSION DU CONTENU DE RHEL 9	33
8.1. SUPPRESSION DES PAQUETS INSTALLÉS	33
8.2. SUPPRESSION DES GROUPES DE PAQUETS	33
8.3. SUPPRESSION DU CONTENU MODULAIRE INSTALLÉ	34
8.4. RESSOURCES SUPPLÉMENTAIRES	40

CHAPITRE 9. GESTION DE L'HISTORIQUE DE LA GESTION DES PAQUETS	41
9.1. TRANSACTIONS D'INSCRIPTION	41
9.2. ANNULATION DES TRANSACTIONS DNF	42
9.3. RÉPÉTITION DES OPÉRATIONS	45
CHAPITRE 10. GESTION DES DÉPÔTS DE LOGICIELS PERSONNALISÉS	46
10.1. DÉFINITION DES OPTIONS DU RÉFÉRENTIEL DNF	46
10.2. AJOUTER UN RÉFÉRENTIEL DNF	46
10.3. ACTIVATION D'UN RÉFÉRENTIEL DNF	47
10.4. DÉSACTIVATION D'UN RÉFÉRENTIEL DNF	47
CHAPITRE 11. GESTION DES VERSIONS DU CONTENU DES FLUX D'APPLICATIONS	49
11.1. DÉPENDANCES MODULAIRES ET CHANGEMENTS DE FLUX	49
11.2. INTERACTION DES DÉPENDANCES MODULAIRES ET NON MODULAIRES	49
11.3. RÉINITIALISATION DES FLUX DE MODULES	50
11.4. DÉSACTIVATION DE TOUS LES FLUX D'UN MODULE	50
ANNEXE A. LISTE DES COMMANDES DNF	51
A.1. COMMANDES POUR LISTER LE CONTENU DANS RHEL 9	51
A.2. COMMANDES POUR L'INSTALLATION DE CONTENU DANS RHEL 9	52
A.3. COMMANDES DE SUPPRESSION DE CONTENU DANS RHEL 9	53

RENDRE L'OPEN SOURCE PLUS INCLUSIF

Red Hat s'engage à remplacer les termes problématiques dans son code, sa documentation et ses propriétés Web. Nous commençons par ces quatre termes : master, slave, blacklist et whitelist. En raison de l'ampleur de cette entreprise, ces changements seront mis en œuvre progressivement au cours de plusieurs versions à venir. Pour plus de détails, voir le [message de notre directeur technique Chris Wright](#).

FOURNIR UN RETOUR D'INFORMATION SUR LA DOCUMENTATION DE RED HAT

Nous apprécions vos commentaires sur notre documentation. Faites-nous savoir comment nous pouvons l'améliorer.

Soumettre des commentaires sur des passages spécifiques

1. Consultez la documentation au format **Multi-page HTML** et assurez-vous que le bouton **Feedback** apparaît dans le coin supérieur droit après le chargement complet de la page.
2. Utilisez votre curseur pour mettre en évidence la partie du texte que vous souhaitez commenter.
3. Cliquez sur le bouton **Add Feedback** qui apparaît près du texte en surbrillance.
4. Ajoutez vos commentaires et cliquez sur **Submit**.

Soumettre des commentaires via Bugzilla (compte requis)

1. Connectez-vous au site Web de [Bugzilla](#).
2. Sélectionnez la version correcte dans le menu **Version**.
3. Saisissez un titre descriptif dans le champ **Summary**.
4. Saisissez votre suggestion d'amélioration dans le champ **Description**. Incluez des liens vers les parties pertinentes de la documentation.
5. Cliquez sur **Submit Bug**.

CHAPITRE 1. OUTILS DE GESTION DES LOGICIELS DANS RED HAT ENTERPRISE LINUX 9

Dans Red Hat Enterprise Linux 9, l'installation du logiciel est assurée par l'outil **DNF** (outil d'installation de logiciels). Red Hat continue de prendre en charge l'utilisation du terme **yum** par souci de cohérence avec les versions majeures précédentes de RHEL. Si vous tapez **yum** au lieu de **dnf**, la commande fonctionne comme prévu car les deux sont des alias pour des raisons de compatibilité.



NOTE

Bien que RHEL 8 et RHEL 9 soient basés sur **DNF**, ils sont compatibles avec **YUM** utilisé dans RHEL 7.

CHAPITRE 2. DISTRIBUTION DU CONTENU DANS RHEL 9

Dans les sections suivantes, vous apprendrez comment le logiciel est distribué dans Red Hat Enterprise Linux 9 :

- Distribution de contenu entre les [référentiels](#).
- [Flux d'applications](#).
- [Modules](#).
- [Flux de modules](#).
- [Profils des modules](#).

2.1. RÉFÉRENTIELS

Le contenu de Red Hat Enterprise Linux 9 est distribué par le biais de deux dépôts principaux : **BaseOS** et **AppStream**. Les ensembles de contenus BaseOS et AppStream sont requis pour une installation RHEL de base et sont disponibles avec tous les abonnements RHEL. Pour les instructions d'installation, voir le document [Exécution d'une installation RHEL 9 standard](#).

BaseOS

Le contenu du référentiel BaseOS consiste en l'ensemble des fonctionnalités du système d'exploitation sous-jacent qui constitue la base de toutes les installations. Ce contenu est disponible au format RPM et est soumis à des conditions d'assistance similaires à celles des versions antérieures de Red Hat Enterprise Linux.

AppStream

Le contenu du référentiel AppStream comprend des applications supplémentaires pour l'espace utilisateur, des langages d'exécution et des bases de données afin de prendre en charge les différentes charges de travail et les différents cas d'utilisation.

CodeReady Linux Builder

Le dépôt CodeReady Linux Builder est disponible avec tous les abonnements RHEL. Il fournit des paquets supplémentaires à l'usage des développeurs. Les paquets inclus dans le dépôt CodeReady Linux Builder ne sont pas pris en charge.

Ressources supplémentaires

- [Effectuer une installation standard de RHEL 9](#)
- [Manifeste de l'emballage](#)

2.2. FLUX D'APPLICATIONS

Les versions multiples des composants de l'espace utilisateur sont fournies sous forme de flux d'applications et mises à jour plus fréquemment que les paquets du système d'exploitation principal. Cela offre une plus grande flexibilité pour personnaliser RHEL sans impacter la stabilité sous-jacente de la plateforme ou des déploiements spécifiques.

Chaque composant Application Stream a un cycle de vie donné, soit identique à celui de RHEL 9, soit plus court, plus adapté à l'application particulière. Pour des informations sur le cycle de vie de RHEL, voir [Red Hat Enterprise Linux Life Cycle](#) et [Red Hat Enterprise Linux Application Streams Life Cycle](#).

Les flux de candidatures sont disponibles dans les formats suivants :

- le format RPM familier
- sous la forme d'une extension du format RPM appelée modules
- en tant que Collections de logiciels
- comme Flatpaks.

RHEL 9 améliore l'expérience des Application Streams en fournissant des versions initiales d'Application Stream qui peuvent être simplement installées en tant que paquets RPM à l'aide de la commande traditionnelle **dnf install**.



NOTE

Certains flux d'applications initiaux au format RPM ont un cycle de vie plus court que Red Hat Enterprise Linux 9.

Certaines versions supplémentaires d'Application Stream seront distribuées sous forme de modules avec un cycle de vie plus court dans les prochaines versions mineures de RHEL 9.

Déterminez toujours la version d'un flux d'applications que vous souhaitez installer et assurez-vous de consulter d'abord le [cycle de vie du flux d'applications de Red Hat Enterprise Linux](#) .



NOTE

Tous les modules ne sont pas des flux d'applications. Les dépendances modulaires ne sont pas considérées comme des flux d'applications.

Ressources supplémentaires

- [Cycle de vie de Red Hat Enterprise Linux](#)
- [Cycle de vie des flux d'applications Red Hat Enterprise Linux](#)
- [Red Hat Enterprise Linux 9 : Guide de compatibilité des applications](#)
- [Manifeste de l'emballage](#)

2.3. MODULES

Un module est un ensemble de paquets RPM qui représentent un composant et sont généralement installés ensemble. Un module typique contient des paquets avec une application, des paquets avec les bibliothèques de dépendances spécifiques à l'application, des paquets avec la documentation de l'application et des paquets avec des utilitaires d'aide.

2.4. FLUX DE MODULES

Les flux de modules sont des filtres qui peuvent être considérés comme des dépôts virtuels dans le dépôt physique d'AppStream. Les flux de modules représentent des versions des composants d'AppStream. Chaque flux reçoit des mises à jour de manière indépendante.

Les flux de modules peuvent être actifs ou inactifs. Les flux actifs donnent au système l'accès aux paquets RPM dans le flux de module particulier, permettant l'installation de la version du composant respectif. Les flux sont actifs s'ils sont explicitement activés par une action de l'utilisateur.

Un seul flux d'un module particulier peut être actif à un moment donné. Par conséquent, une seule version d'un composant peut être installée sur un système. Des versions différentes peuvent être utilisées dans des conteneurs distincts.

Chaque module peut avoir un flux par défaut. Dans RHEL 9, aucun flux par défaut n'est défini, mais vous pouvez configurer le vôtre comme décrit dans la section [Configuration de flux et de profils de modules personnalisés par défaut](#).

Certains flux de modules peuvent dépendre d'autres flux de modules.

Pour sélectionner un flux particulier pour une application utilisateur d'exécution ou une application développeur, considérez ce qui suit :

- Fonctionnalité requise et versions des composants prenant en charge cette fonctionnalité
- Compatibilité
- Durée du [cycle de vie](#) et plan de mise à jour

Pour une liste de tous les modules et flux disponibles, voir le [manifeste du paquet](#). Pour connaître les modifications apportées à chaque composant, voir les [notes de mise à jour](#).

Ressources supplémentaires

- [Dépendances modulaires et changements de flux](#)

2.5. PROFILS DES MODULES

Un profil est une liste de paquets recommandés à installer ensemble pour un cas d'utilisation particulier tel que serveur, client, développement, installation minimale, ou autre. Ces listes de paquets peuvent contenir des paquets extérieurs au flux de modules, généralement issus du référentiel BaseOS ou des dépendances du flux.

L'installation de paquets à l'aide d'un profil est une action unique fournie pour la commodité de l'utilisateur. Elle n'empêche pas l'installation ou la désinstallation des paquets fournis par le module. Il est également possible d'installer des paquets en utilisant plusieurs profils du même flux de modules sans aucune autre étape préparatoire.

Chaque flux de modules peut avoir un nombre quelconque de profils, y compris aucun. Pour un flux de modules donné, certains de ses profils peuvent être marqués comme *default* et sont alors utilisés pour les actions d'installation de profil lorsqu'aucun profil n'est explicitement spécifié. Toutefois, l'existence d'un profil par défaut pour un flux de modules n'est pas obligatoire.

Exemple 2.1. nodejs profils des modules

Le module **nodejs**, qui fournit l'environnement d'exécution **Node.js**, propose les profils d'installation suivants :

- **common** - les paquets prêts à la production.
Il s'agit du profil par défaut.
- **development** - nécessaires pour apporter des modifications à **Node.js**.

- **minimal** - le plus petit ensemble de paquets qui fournit l'environnement d'exécution **Node.js**.
- **s2i** - nécessaires à la création de conteneurs Linux **Node.js** Source-to-Image (S2I).

CHAPITRE 3. CONFIGURATION DE LA DNF

Les informations de configuration de **DNF** et des utilitaires associés sont stockées dans le fichier `/etc/dnf/dnf.conf`. Ce fichier contient une section obligatoire **[main]**, qui vous permet de définir des options ayant un effet global **DNF** options qui ont un effet global.

Dans les sections suivantes, vous apprendrez à configurer **DNF** à l'aide du fichier de configuration `/etc/dnf/dnf.conf`:

- Visualiser les configurations actuelles de **DNF** .
- Définir les options de **DNF [main]** .
- Utiliser **DNF** des plug-ins.

3.1. VISUALISATION DES CONFIGURATIONS DNF ACTUELLES

Pour afficher la configuration actuelle **DNF** la configuration actuelle, procédez comme suit.

Procédure

- Pour afficher les valeurs actuelles des options globales **DNF** spécifiées dans la section **[main]** du fichier `/etc/dnf/dnf.conf`, utilisez :

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

3.2. RÉGLAGE DES OPTIONS PRINCIPALES DE DNF

Le fichier de configuration `/etc/dnf/dnf.conf` contient une section **[main]**. Les paires clé-valeur de cette section affectent le fonctionnement et le traitement des dépôts de **DNF** fonctionne et traite les référentiels.

Vous pouvez ajouter des options supplémentaires sous l'intitulé de la section **[main]** dans `/etc/dnf/dnf.conf`.

Pour une liste complète des options disponibles sur **[main]**, voir la section **[main] OPTIONS** de la page de manuel `dnf.conf(5)`.

3.3. UTILISATION DES PLUG-INS DNF

DNF fournit des plug-ins qui étendent et améliorent son fonctionnement. Certains plug-ins sont installés par défaut.

3.3.1. Gestion des plug-ins DNF

Les fichiers de configuration du plug-in contiennent toujours une section **[main]** dans laquelle l'option **enabled=** détermine si le plug-in est activé lorsque vous exécutez des commandes **dnf**. Si cette option est manquante, vous pouvez l'ajouter manuellement au fichier.

Chaque plug-in installé peut avoir son propre fichier de configuration dans le répertoire `/etc/dnf/plugins/`. Ces fichiers permettent d'activer ou de désactiver des options spécifiques au plug-in.

3.3.2. Activation et désactivation des plug-ins DNF

Dans l'outil **DNF** les plug-ins sont chargés par défaut.

Pour modifier le chargement des **DNF** et activer ou désactiver des **DNF** spécifiques, procédez comme suit.

Procédure

- Pour désactiver ou activer le chargement des **DNF** assurez-vous qu'une ligne commençant par **plugins=** est présente dans la section **[main]** du fichier **/etc/dnf/dnf.conf**.
 1. Pour désactiver le chargement des **DNF** pour désactiver le chargement des plug-ins, réglez la valeur de **plugins=** sur **0**.



IMPORTANT

Il est conseillé de désactiver tous les plug-ins sur le site **not**. Certains plug-ins fournissent des services **DNF** importants. En particulier, les plug-ins **product-id** et **subscription-manager** assurent la prise en charge du système de certificats **Content Delivery Network (CDN)**. La désactivation globale des modules d'extension est fournie à titre d'option de commodité et n'est conseillée que pour diagnostiquer un problème potentiel avec le logiciel **DNF**.

2. Pour permettre le chargement des **DNF** les plug-ins, définissez la valeur de **plugins=** sur **1**.
- Pour désactiver tous les **DNF** pour une commande particulière, ajoutez l'option **--noplugins** à la commande. Par exemple, pour désactiver **DNF** les plug-ins pour la commande **update**:

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

- Pour désactiver certains **DNF** pour une seule commande, ajoutez l'option **--disableplugin=plugin-name** à la commande. Par exemple, pour désactiver certains **DNF** pour la commande **update**:

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

Remplacez *plugin-name* par le nom du plug-in.

- Pour activer certains **DNF** pour une seule commande, ajoutez l'option **--enableplugin=plugin-name** à la commande. Par exemple, pour activer certains **DNF** pour la commande **update**:

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

Remplacez *plugin-name* par le nom du plug-in.

CHAPITRE 4. RECHERCHE DE CONTENU SUR RHEL 9

Dans les sections suivantes, vous apprendrez comment localiser et examiner le contenu des dépôts AppStream et BaseOS dans Red Hat Enterprise Linux 9 à l'aide de **DNF**:

- [Recherche de paquets](#) fournissant le contenu souhaité.
- [Liste des paquets installés et disponibles](#).
- [Liste des dépôts activés et désactivés](#).
- [Affiche des informations sur les paquets installés et disponibles](#).
- [Liste des groupes de paquets installés et disponibles](#).
- [Dresser la liste des modules disponibles et obtenir des informations détaillées à leur sujet](#).
- [Spécifier des expressions globales dans l'entrée **dnf**](#).

4.1. RECHERCHE DE PROGIIELS

Pour trouver un paquet qui fournit une application particulière ou un autre contenu, suivez les étapes suivantes.

Procédure

- Pour rechercher un paquet, utilisez :

```
$ dnf search term
```

Remplacer *term* par un terme relatif au paquet.

Notez que la commande **dnf search** renvoie les termes correspondant au nom et au résumé des paquets. La recherche est ainsi plus rapide et vous pouvez rechercher des paquets dont vous ne connaissez pas le nom, mais pour lesquels vous connaissez un terme apparenté.

- Pour inclure des correspondances de termes dans les descriptions de paquets, utilisez :

```
$ dnf search --all term
```

Remplacez *term* par un terme que vous souhaitez rechercher dans le nom, le résumé ou la description d'un paquet.

Notez que la commande **dnf search --all** permet une recherche plus exhaustive mais plus lente.

- Pour rechercher un paquet et sa version ou un fichier :

```
$ dnf repoquery name
```

Remplacez *name* par le nom du paquet ou du fichier que vous souhaitez rechercher.

- Recherche d'un paquetage, d'un binaire ou d'un fichier :

```
$ dnf provides name
```

Remplacez *name* par le nom d'un paquetage, d'un binaire ou d'un fichier que vous souhaitez rechercher.

4.2. LISTER LES PROGICIELS

Pour dresser la liste des paquets installés et des paquets disponibles pour l'installation, procédez comme suit.

Procédure

- Pour obtenir la liste des informations sur tous les paquets installés et disponibles, utilisez :

```
$ dnf list --all
```

- Pour obtenir la liste de tous les paquets installés sur votre système, utilisez l'option :

```
$ dnf list --installed
```

Alternativement :

```
$ dnf repoquery --installed
```

- Pour obtenir la liste de tous les paquets disponibles à l'installation dans tous les dépôts activés, utilisez :

```
$ dnf list --available
```

Alternativement :

```
$ dnf repoquery
```



NOTE

Vous pouvez filtrer les résultats en ajoutant des expressions globales en tant qu'arguments. Pour plus de détails, voir [Spécifier des expressions globales dans l'entrée dnf](#).

4.3. LISTER LES RÉFÉRENTIELS

Pour dresser la liste des référentiels activés et désactivés sur votre système, procédez comme suit.

Procédure

- Pour lister tous les dépôts activés sur votre système, utilisez :

```
$ dnf repolist
```

- Pour lister tous les dépôts désactivés sur votre système, utilisez :

```
$ dnf repolist --disabled
```

- Pour lister les dépôts activés et désactivés, utilisez :

```
$ dnf replist --all
```

- Pour lister des informations supplémentaires sur les dépôts, utilisez :

```
$ dnf reponfo
```



NOTE

Vous pouvez filtrer les résultats en passant l'ID ou le nom des dépôts comme arguments ou en ajoutant des expressions globales. Pour plus de détails, voir [Spécifier des expressions globales dans l'entrée `dnf`](#).

4.4. AFFICHAGE DES INFORMATIONS SUR LE PAQUET

Pour afficher des informations sur les paquets disponibles ou les paquets installés sur votre système, procédez comme suit.

Procédure

- Pour afficher des informations sur un ou plusieurs paquets disponibles, utilisez :

```
$ dnf info package-name
```

Remplacez *package-name* par le nom du paquet.

Alternativement :

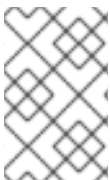
```
$ dnf repoquery --info package-name
```

Remplacez *package-name* par le nom du paquet.

- Pour afficher des informations sur un ou plusieurs paquets installés sur votre système, utilisez :

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

Remplacez *package-name* par le nom du paquet.



NOTE

Vous pouvez filtrer les résultats en ajoutant des expressions globales en tant qu'arguments. Pour plus de détails, voir [Spécifier des expressions globales dans l'entrée `dnf`](#).

4.5. LISTER LES GROUPES DE PAQUETS

Pour dresser la liste des groupes de paquets installés sur votre système ou des groupes de paquets disponibles pour installation, procédez comme suit.

Procédure

- Pour afficher le nombre de groupes installés et disponibles, utilisez :

```
$ dnf group summary
```

- Pour obtenir la liste de tous les groupes installés et disponibles, utilisez l'option :

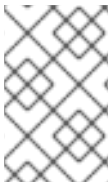
```
$ dnf group list
```

Notez que vous pouvez filtrer les résultats en ajoutant des options de ligne de commande pour la commande **dnf group list** (**--hidden**, **--available**). Pour plus d'options disponibles, voir les pages de manuel.

- Pour dresser la liste des paquets obligatoires et optionnels contenus dans un groupe particulier, utilisez :

```
$ dnf group info group-name
```

Remplacez *group-name* par le nom du groupe.



NOTE

Vous pouvez filtrer les résultats en ajoutant des expressions globales en tant qu'arguments. Pour plus de détails, voir [Spécifier des expressions globales dans l'entrée dnf](#).

4.6. LISTE DES MODULES DISPONIBLES ET DE LEUR CONTENU

Pour connaître les modules disponibles et leurs caractéristiques, suivez les étapes suivantes.

Procédure

- Pour dresser la liste des flux de modules disponibles pour votre système :

```
$ dnf module list
```

La sortie de cette commande répertorie les flux de modules avec le nom, le flux, les profils et le résumé sur une ligne séparée.

- Pour afficher les détails d'un module, y compris une description, une liste de tous les profils et une liste de tous les paquets fournis :

```
$ dnf module info module-name
```

- Pour dresser la liste des paquets installés par chacun des profils de module :

```
$ dnf module info --profile module-name
```

- Pour afficher l'état actuel d'un module, y compris les flux activés et les profils installés :

```
$ dnf module list module-name
```

- Pour savoir quels modules, flux et profils fournissent un paquet spécifique :

■

\$ dnf module provides *package*

Si le paquet est disponible en dehors de tout module, la sortie de cette commande est vide.

Exemple 4.1. Obtenir des informations sur un module

Voici un exemple de la façon de dresser la liste des modules disponibles et d'obtenir des informations sur le contenu du module **nodejs**.

**NOTE**

Les résultats de cet exemple ont été édités pour des raisons de concision. Les sorties réelles peuvent contenir plus d'informations que celles présentées ici.

1. Liste des modules disponibles :

```
$ dnf module list
(...)
rhel-AppStream
Name      Stream Profiles  Summary
maven     3.8   common [d]  Java project management and project comprehension tool
nodejs    18    common [d], Javascript runtime
           development,
           minimal,
           s2i
php       8.1   common [d], PHP scripting language
           devel,
           minimal
ruby      3.1   common [d]  An interpreter of object-oriented scripting language

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

2. Examinez les détails du module **nodejs**:

```
$ dnf module info nodejs

Name       : nodejs
Stream     : 18
Version    : 9010020220808155010
Context    : rhel9
Architecture : x86_64
Profiles   : common [d], development, minimal, s2i
Default profiles : common
Repo       : rhel-AppStream
Summary    : Javascript runtime
Description : (...)
(...)

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

Si vous ne spécifiez aucun flux, **dnf** répertorie tous les flux disponibles.

3. Examiner les profils disponibles dans la filière **18** du module **nodejs**:

■

```
$ dnf module info --profile nodejs:18
(...)
Name      : nodejs:18:9010020220808155010:rhel9:x86_64
common    : nodejs
           : npm
development : nodejs
           : nodejs-devel
           : npm
minimal    : nodejs
s2i       : nodejs
           : nodejs-nodemon
           : npm
```

Notez que chaque profil installe un ensemble différent de paquets, y compris leurs dépendances.

4. Installez le flux du module **nodejs:18** avec son profil **common**:

```
# dnf module install nodejs:18/common
(...)
Dependencies resolved.
=====
Package      Architecture Version
Repository   Size
=====
Installing group/module packages:
nodejs       x86_64      1:18.7.0-1.module+el9.1.0+16284+4fdefb2f
             rhel-AppStream 12 M
npm          x86_64      1:8.15.0-1.18.7.0.1.module+el9.1.0+16284+4fdefb2f
             rhel-AppStream 2.2 M
Installing weak dependencies:
nodejs-docs  noarch      1:18.7.0-1.module+el9.1.0+16284+4fdefb2f
             rhel-AppStream 7.2 M
nodejs-full-i18n x86_64      1:18.7.0-1.module+el9.1.0+16284+4fdefb2f
             rhel-AppStream 8.2 M
Installing module profiles:
nodejs/common
Enabling module streams:
nodejs              18

Transaction Summary
=====
Install 4 Packages

Total download size: 30 M
Installed size: 155 M
Is this ok [y/N]: y
(...)
```

5. Inspecter l'état actuel du module **nodejs**:

```
$ dnf module list nodejs
```

```
(...)
Name      Stream  Profiles      Summary
nodejs    18 [e]  common [d] [i],
          development,
          minimal,
          s2i
```

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

La sortie montre que le flux **nodejs:18** est activé et que son profil **common** est installé.

Ressources supplémentaires

- [Modules](#)
- [Flux de modules](#)
- [Profils des modules](#)

4.7. SPÉCIFICATION D'EXPRESSIONS GLOBALES DANS L'ENTRÉE DNF

Avec les commandes **dnf**, vous pouvez filtrer les résultats en ajoutant un ou plusieurs *global expressions* comme arguments. Les expressions globales doivent être échappées lorsqu'elles sont transmises en tant qu'arguments à la commande **dnf**.

Procédure

- Pour vous assurer que les expressions globales sont transmises à **dnf** comme prévu, utilisez l'une des méthodes suivantes :
 - Mettez l'expression globale entière entre guillemets simples ou doubles :

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

Remplacez *file-name* par le nom du fichier.

Notez que le *file-name* doit être précédé d'une séquence de caractères */* ou **/* pour obtenir le résultat souhaité.

- Échapper aux caractères génériques en les faisant précéder d'une barre oblique inverse (`\`) :

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

Remplacez *file-name* par le nom du fichier.

4.8. RESSOURCES SUPPLÉMENTAIRES

- [Commandes pour lister le contenu dans RHEL 9](#)

CHAPITRE 5. INSTALLATION DU CONTENU DE RHEL 9

Dans les sections suivantes, vous apprendrez à installer le contenu dans Red Hat Enterprise Linux 9 :

- Installez un [paquetage](#), un [groupe de paquets](#) ou des [ensembles de paquets](#) fournis par des modules, des flux et des profils à l'aide de la commande **dnf**.
- [Exécutez le contenu installé de RHEL 9](#) .
- [Configurer des flux et des profils de modules personnalisés par défaut](#) .

5.1. INSTALLATION DES PAQUETS

Pour installer des paquets à l'aide de la commande **dnf install**, procédez comme suit.

Procédure

- Pour installer un seul paquet, utilisez :

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

Remplacez *package-name* par le nom du paquet.

- Si le paquetage n'est fourni par aucun flux de modules, cette procédure est identique à celle utilisée sur les versions antérieures de Red Hat Enterprise Linux.
 - Si le paquet est fourni par un flux de modules qui est activé, le paquet est installé sans autre manipulation.
 - Si le paquet est fourni par un flux de modules qui n'est pas activé, vous devez activer manuellement le flux de modules concerné avant d'installer le paquet.
- Pour installer plusieurs paquets et leurs dépendances simultanément, utilisez :

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

Remplacez *package-name-1* et *package-name-2* par les noms des paquets.

- Lorsque vous installez des paquets sur un système *multilib* (AMD64, machine Intel 64), vous pouvez spécifier l'architecture du paquet en l'ajoutant au nom du paquet :

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

Remplacez *package-name.arch* par le nom et l'architecture du paquet.

- Si vous connaissez le nom du binaire que vous souhaitez installer, mais pas le nom du paquet, vous pouvez utiliser le chemin d'accès au binaire comme argument :

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

Remplacer */usr/sbin/binary-file* par le chemin d'accès au fichier binaire.

dnf effectue une recherche dans les listes de paquets, trouve le paquet qui fournit */usr/sbin/binary-file* et vous demande si vous souhaitez l'installer.

- Pour installer un paquetage précédemment téléchargé à partir d'un répertoire local, utilisez :

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

Remplacez */path/* par le chemin d'accès au paquet.

Ressources supplémentaires

- [Installer un contenu modulaire](#)

5.2. INSTALLATION DE GROUPES DE PAQUETS

Pour installer un groupe de paquets à partir d'un nom de groupe ou d'un identifiant de groupe à l'aide de la commande **dnf group install**, procédez comme suit.

Procédure

- Pour installer un groupe de paquets à partir d'un nom de groupe, utilisez :

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

Remplacez *group-name* par le nom complet du groupe ou du groupe environnemental.

- Pour installer un groupe de paquets à partir d'un identifiant de groupe, utilisez :

```
# dnf group install groupID
```

Remplacez *groupID* par l'ID du groupe.

5.3. INSTALLER UN CONTENU MODULAIRE

Dans RHEL 9, aucun flux de module par défaut n'est prédéfini. Vous devez activer le flux requis avant d'installer un paquet.

Pour installer le contenu modulaire fourni par un flux de modules ou un profil à l'aide de la commande **dnf module install**, procédez comme suit.

Conditions préalables

- Vous n'avez pas de paquets installés à partir d'un autre flux du même module.

Procédure

- Pour installer un flux de modules sélectionné, utilisez

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

Le flux sélectionné est automatiquement activé. Si un profil par défaut est défini pour le flux, ce profil est automatiquement installé.

Alternativement :

1. Activer un flux de modules :

dnf module enable *module-name:stream*

Remplacez *module-name* et *stream* par les noms du module et du flux. Le module stream est maintenant activé mais aucun paquet n'est installé.

2. Installez un flux de modules actifs que vous avez activé :

dnf module install *module-name*

- Pour installer un profil sélectionné du flux de modules, utiliser :

dnf module install *module-name:stream/profile*

Cela permet d'activer le flux et d'installer l'ensemble des paquets recommandés pour un flux (version) et un profil (objectif) donnés du module.

Exemple 5.1. Installation d'un flux d'une application

Voici un exemple d'installation d'une application à partir d'un flux spécifique, à savoir **Node.js** dans la version **18**.

1. Liste des modules qui fournissent le paquetage **nodejs** pour voir quels flux sont disponibles :

```
$ dnf module list nodejs
(...)
rhel-AppStream
Name      Stream Profiles      Summary
nodejs    18    common [d],      Javascript runtime
          development,
          minimal,
          s2i
```

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

La sortie montre que le module **nodejs** est disponible avec le flux **18**.

2. Installer les paquets fournis par le module **nodejs** dans le flux **18**:

```
# dnf module install nodejs:18
(...)
Dependencies resolved.
=====
Package      Architecture  Version
Repository   Size
=====
Installing group/module packages:
nodejs       x86_64        1:18.7.0-1.module+el9.1.0+16284+4fdefb2f
rhel-AppStream 12 M
npm          x86_64        1:8.15.0-1.18.7.0.1.module+el9.1.0+16284+4fdefb2f
rhel-AppStream 2.2 M
Installing weak dependencies:
nodejs-docs  noarch        1:18.7.0-1.module+el9.1.0+16284+4fdefb2f
rhel-AppStream 7.2 M
```

```

nodejs-full-i18n x86_64      1:18.7.0-1.module+el9.1.0+16284+4fdefb2f
rhel-AppStream   8.2 M
Installing module profiles:
nodejs/common
Enabling module streams:
nodejs                18

Transaction Summary
=====
=====
Install 4 Packages

Total download size: 30 M
Installed size: 155 M
Is this ok [y/N]: y
(...)
Complete!

```

Le profil d'installation n'ayant pas été spécifié, le profil par défaut **common** a été utilisé.

3. Vérifiez la version installée de **Node.js**:

```

$ node -v
v18.7.0

```

Ressources supplémentaires

- [Modules](#)
- [Flux de modules](#)
- [Profils des modules](#)

5.4. EXÉCUTION DU CONTENU INSTALLÉ

Les nouvelles commandes sont généralement activées après l'installation du contenu des dépôts RHEL 9. Si les commandes proviennent d'un paquet RPM ou si les paquets RPM ont été activés par un module, l'expérience d'utilisation de la commande ne devrait pas être différente.

Procédure

- Pour exécuter les nouvelles commandes, saisissez-les directement :

```

$ command

```

Remplacez *command* par le nom de la commande que vous souhaitez exécuter.



NOTE

Dans RHEL 9, GCC Toolset est présenté sous la forme d'une collection de logiciels. Pour exécuter une commande à partir d'un composant empaqueté en tant que collection de logiciels, utilisez :

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

Remplacez *collection* par le nom de la collection de logiciels.

Pour plus d'informations, voir [Utilisation de la boîte à outils GCC](#) .

5.5. CONFIGURATION DE FLUX ET DE PROFILS DE MODULES PERSONNALISÉS PAR DÉFAUT

Dans RHEL 9, aucun flux par défaut n'est défini dans le référentiel qui contient les modules. Vous pouvez configurer un flux et un profil par défaut en créant un fichier de configuration dans le répertoire **/etc/dnf/modules.defaults.d/**.

Lorsque vous définissez un flux personnalisé par défaut, vous pouvez installer des paquets à partir de ce flux sans activer le flux au préalable et sans spécifier le flux lors de l'installation du module.

Lorsque vous définissez un profil personnalisé par défaut, ce profil sera automatiquement choisi lorsque vous installerez le flux de modules pour lequel il est défini.

Pour configurer le flux et le profil par défaut via le répertoire **/etc/dnf/modules.defaults.d/**, procédez comme suit.

Conditions préalables

- Vous comprenez le [concept](#) de [active module stream](#).

Procédure

- Créez un fichier de configuration YAML dans le répertoire de dépôt **/etc/dnf/modules.defaults.d/**.

```
document: modulemd-defaults
version: 1
data:
  module: <module>
  stream: "<stream>"
  profiles:
    '<stream>': [<profile>]
    '<stream>': [<profile>]
  ...
```

La sortie précédente représente la définition par défaut présente pour le module **<module>** au moment de la rédaction de ce document. Voir la [spécification en amont](#) du format de métadonnées modulaire.

Exemple 5.2. Configuration du flux du module `nodejs:18` comme flux par défaut et modification du profil par défaut

L'exemple suivant montre comment configurer le flux **18** du module **nodejs** comme flux par défaut et le profil **minimal** comme profil par défaut.

1. Examinez le module **nodejs**:

```
# dnf module list nodejs
(...)
rhel-AppStream
Name      Stream    Profiles      Summary
nodejs    18        common [d],   Javascript runtime
           development,
           minimal,
           s2i
```

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

Actuellement, le flux du module **nodejs:18** n'est pas défini comme flux par défaut et le profil par défaut est le profil **common**.

2. Configurez le flux par défaut sur **18** et le profil par défaut sur **minimal** en implémentant la configuration suivante dans le fichier YAML du répertoire **/etc/dnf/modules.defaults.d/**:

```
document: modulemd-defaults
version: 1
data:
  module: nodejs
  stream: "18"
  profiles:
    '18': [minimal]
```

3. Examinez à nouveau le module **nodejs**:

```
# dnf module list nodejs
(...)
rhel-AppStream
Name      Stream    Profiles      Summary
nodejs    18 [d]     common,       Javascript runtime
           development,
           minimal [d],
           s2i
```

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

Le flux du module **nodejs:18** est maintenant défini comme flux par défaut et le profil **minimal** comme profil par défaut de ce flux.

5.6. RESSOURCES SUPPLÉMENTAIRES

- [dnf\(8\)](#) page de manuel
- [Commandes pour l'installation de contenu dans RHEL 9](#)
- [Définition du fichier YAML de métadonnées modulemd](#)

CHAPITRE 6. MISE À JOUR DU CONTENU DE RHEL 9

Avec **DNF** vous pouvez vérifier si votre système a des mises à jour en attente. Vous pouvez dresser la liste des paquets à mettre à jour et choisir de mettre à jour un seul paquet, plusieurs paquets ou tous les paquets à la fois. Si l'un des paquets que vous choisissez de mettre à jour a des dépendances, il est également mis à jour.

Dans les sections suivantes, vous apprendrez comment mettre à jour le contenu de Red Hat Enterprise Linux 9 à l'aide de la commande **DNF**:

- [Vérifier les mises à jour disponibles](#) .
- [Mise à jour d'un seul paquet, d'un groupe de paquets ou de tous les paquets et de leurs dépendances](#).
- [Appliquer les mises à jour de sécurité](#) .

6.1. VÉRIFICATION DES MISES À JOUR

Pour vérifier les mises à jour disponibles pour les paquets installés sur votre système à l'aide de la commande **dnf check-update**, procédez comme suit.

Procédure

- Exécutez la commande suivante pour voir quels paquets installés sur votre système ont des mises à jour disponibles :

```
# dnf check-update
```

La sortie renvoie la liste des paquets et de leurs dépendances pour lesquels une mise à jour est disponible.

6.2. MISE À JOUR DES PAQUETS

Pour mettre à jour un seul paquet, un groupe de paquets ou tous les paquets et leurs dépendances à l'aide de la commande **dnf upgrade**, procédez comme suit.

Procédure

- Pour mettre à jour tous les paquets et leurs dépendances, utilisez :

```
# dnf upgrade
```

- Pour mettre à jour un seul paquet, utilisez :

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

Remplacez *package-name* par le nom du paquet.

- Pour mettre à jour un groupe de paquets, utilisez

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

Remplacez *group-name* par le nom du groupe de paquets.



IMPORTANT

Lorsque vous appliquez des mises à jour au noyau, **dnf** installe toujours un nouveau noyau, que vous utilisiez la commande **dnf upgrade** ou **dnf install**.

6.3. MISE À JOUR DES PAQUETS LIÉS À LA SÉCURITÉ

Pour mettre à jour les paquets liés à la sécurité à l'aide de la commande **dnf upgrade** ou **dnf upgrade-minimal**, procédez comme suit.

Procédure

- Pour mettre à jour vers les derniers paquets disponibles qui ont des errata de sécurité, utilisez :

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

- Pour mettre à jour vers les derniers paquets de sécurité errata, utilisez :

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

CHAPITRE 7. AUTOMATISATION DES MISES À JOUR LOGICIELLES DANS RHEL 9

Pour vérifier et télécharger automatiquement et régulièrement les mises à jour des paquets, vous pouvez utiliser l'outil **DNF Automatic** fourni par le paquetage **dnf-automatic**.

DNF Automatic est une interface de ligne de commande alternative à **DNF** qui est adaptée à une exécution automatique et régulière à l'aide de minuteries **systemd**, de tâches **cron** et d'autres outils de ce type.

DNF Automatic synchronise les métadonnées des paquets si nécessaire, vérifie si des mises à jour sont disponibles, puis effectue l'une des actions suivantes en fonction de la configuration de l'outil :

- Sortie
- Télécharger les paquets mis à jour
- Télécharger et appliquer les mises à jour

Le résultat de l'opération est ensuite communiqué par un mécanisme choisi, tel que la sortie standard ou le courrier électronique.

Dans les sections suivantes, vous apprendrez comment automatiser les mises à jour logicielles dans Red Hat Enterprise Linux 9 :

- [Installer l'outil **DNF Automatic**](#) .
- [Activer l'outil **DNF Automatic**](#) .
- [Examinez le fichier de configuration **DNF Automatic** et ses sections](#) .
- [Examinez les unités de temporisation de **dnf-automatic** **systemd**](#) .

7.1. INSTALLATION DE DNF AUTOMATIC

Pour installer l'outil **DNF Automatic** sur votre système, suivez les étapes suivantes.

Procédure

- Installez le paquetage **dnf-automatic**:

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

Vérification

- Vérifiez la réussite de l'installation en confirmant la présence du paquet **dnf-automatic**:

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

7.2. DNF FICHER DE CONFIGURATION AUTOMATIQUE

Par défaut, **DNF Automatic** utilise **/etc/dnf/automatic.conf** comme fichier de configuration pour définir son comportement.

Le fichier de configuration est divisé en plusieurs sections thématiques :

- **[commands]** section
Définit le mode de fonctionnement de **DNF Automatic**.
- **[emitters]** section
Définit la manière dont les résultats de **DNF Automatic** sont rapportés.
- **[command_email]** section
Fournit la configuration de l'émetteur de courrier électronique pour une commande externe utilisée pour envoyer des courriers électroniques.
- **[email]** section
Fournit la configuration de l'émetteur de courrier électronique.
- **[base]** section
Remplace les paramètres du fichier de configuration principal de **DNF**.

Avec les paramètres par défaut du fichier **/etc/dnf/automatic.conf**, **DNF Automatic** recherche les mises à jour disponibles, les télécharge et affiche les résultats sur la sortie standard.



AVERTISSEMENT

Les paramètres du mode de fonctionnement de la section **[commands]** sont remplacés par les paramètres utilisés par une unité de temporisation **systemd** pour toutes les unités de temporisation à l'exception de **dnf-automatic.timer**.

Ressources supplémentaires

- [DNF Documentation automatique](#)
- **dnf-automatic(8)** page de manuel
- [Vue d'ensemble des unités de temporisation systemd incluses dans le paquetage dnf-automatic](#)

7.3. ACTIVATION DU DNF AUTOMATIQUE

Pour exécuter **DNF Automatic** vous devez toujours activer et démarrer une unité de temporisation **systemd** spécifique. Vous pouvez utiliser l'une des unités de temporisation fournies dans le paquetage **dnf-automatic**, ou vous pouvez écrire votre propre unité de temporisation en fonction de vos besoins.

Pour activer **DNF Automatic** sur votre système, suivez les étapes suivantes.

Conditions préalables

- Vous avez spécifié le comportement de **DNF Automatic** en modifiant le fichier de configuration de **/etc/dnf/automatic.conf**.

Procédure

- Pour sélectionner, activer et démarrer une unité de temporisation systemd qui **downloads** les mises à jour disponibles, utilisez :

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

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

- Pour sélectionner, activer et démarrer une unité de temporisation systemd qui **downloads and installs** les mises à jour disponibles, utilisez :

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

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

- Pour sélectionner, activer et démarrer une unité de temporisation systemd qui **reports** les mises à jour disponibles, utilisez :

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

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

- Pour sélectionner, activer et démarrer une unité de temporisation systemd dont les mises à jour disponibles sont **downloads**, **downloads and installs**, ou **reports**, utilisez :

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

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

- Il est possible de sélectionner, d'activer et de démarrer une unité de temporisation systemd en une seule commande en utilisant l'option **--now**. Par exemple :

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



NOTE

Vous pouvez également exécuter **DNF Automatic** en exécutant le fichier **/usr/bin/dnf-automatic** directement à partir de la ligne de commande ou d'un script personnalisé.

Vérification

- Verify that the timer is enabled:

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

Ressources supplémentaires

- **dnf-automatic(8)** page de manuel
- [Vue d'ensemble des unités de temporisation systemd incluses dans le paquetage dnf-automatic](#)
- [DNF Fichier de configuration automatique](#)

7.4. VUE D'ENSEMBLE DES UNITÉS TIMER DE SYSTEMD INCLUSES DANS LE PAQUETAGE DNF-AUTOMATIC

Les unités de temporisation de systemd sont prioritaires et remplacent les paramètres du fichier de configuration `/etc/dnf/automatic.conf` lors du téléchargement et de l'application des mises à jour.

Par exemple, si vous avez défini `download_updates = yes` dans le fichier de configuration `/etc/dnf/automatic.conf`, mais que vous avez activé `dnf-automatic-notifyonly.timer unit`, les paquets ne seront pas téléchargés.

Le paquetage `dnf-automatic` comprend les unités de timer systemd suivantes :

Tableau 7.1. timers systemd inclus dans le paquetage `dnf-automatic`

Unité de minuterie	Fonction	Remplace les paramètres du fichier <code>/etc/dnf/automatic.conf</code> ?
<code>dnf-automatic-download.timer</code>	Télécharge les paquets dans le cache et les rend disponibles pour la mise à jour. Remarque : cette unité de minuterie n'installe pas les paquets mis à jour. Pour effectuer l'installation, vous devez exécuter la commande <code>dnf update</code> .	Oui
<code>dnf-automatic-install.timer</code>	Télécharge et installe les paquets mis à jour.	Oui
<code>dnf-automatic-notifyonly.timer</code>	Télécharge uniquement les données du référentiel pour maintenir le cache du référentiel à jour et vous informe des mises à jour disponibles. Remarque : cette unité de minuterie ne télécharge ni n'installe les paquets mis à jour.	Oui
<code>dnf-automatic.timer</code>	Le comportement de ce temporisateur lors du téléchargement et de l'application des mises à jour est spécifié par les paramètres du fichier de configuration <code>/etc/dnf/automatic.conf</code> . Le comportement par défaut est le même que pour l'unité <code>dnf-automatic-download.timer</code> : elle télécharge les paquets, mais ne les installe pas.	Non

Ressources supplémentaires

- **dnf-automatic(8)** page de manuel
- [DNF Fichier de configuration automatique](#)

CHAPITRE 8. SUPPRESSION DU CONTENU DE RHEL 9

Dans les sections suivantes, vous apprendrez comment supprimer du contenu dans Red Hat Enterprise Linux 9 à l'aide de la commande **DNF**:

- [Retirer un paquet.](#)
- [Supprimer un groupe de paquets.](#)
- [Supprimer le contenu installé dans un flux de modules ou un profil.](#)
 - [Supprimer les paquets d'un profil sélectionné.](#)
 - [Supprime tous les paquets d'un flux de modules sélectionné.](#)

8.1. SUPPRESSION DES PAQUETS INSTALLÉS

Pour supprimer les paquets installés sur votre système à l'aide de la commande **dnf remove**, procédez comme suit.

Procédure

- Pour supprimer un paquet particulier et tous les paquets dépendants inutilisés, utilisez :

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

Remplacez *package-name* par le nom du paquet.

- Pour supprimer simultanément plusieurs paquets et leurs dépendances inutilisées, utilisez :

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

Remplacez *package-name-1* et *package-name-2* par les noms des paquets.



NOTE

La commande **dnf** supprime un paquet ainsi que tous les autres paquets dépendants.

8.2. SUPPRESSION DES GROUPES DE PAQUETS

Pour supprimer un groupe de paquets par le nom du groupe ou l'identifiant du groupe à l'aide de la commande **dnf group remove**, procédez comme suit.

Procédure

- Pour supprimer un groupe de paquets à partir du nom du groupe, utilisez :

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

Remplacez *group-name* par le nom complet du groupe.

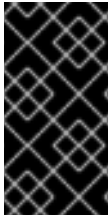
- Pour supprimer un groupe de paquets en fonction de l'identifiant du groupe, utilisez :

```
# dnf group remove groupID
```

Remplacez *groupID* par l'ID du groupe.

8.3. SUPPRESSION DU CONTENU MODULAIRE INSTALLÉ

Lors de la suppression du contenu modulaire installé, vous pouvez supprimer les paquets d'un [profil sélectionné](#) ou de [l'ensemble du flux](#).



IMPORTANT

DNF essaiera de supprimer tous les paquets dont le nom correspond aux paquets installés avec un profil ou un flux, y compris les paquets dépendants. Vérifiez toujours la liste des paquets à supprimer avant de procéder, en particulier si vous avez activé les dépôts personnalisés sur votre système.

8.3.1. Suppression des paquets d'un profil installé

Lorsque vous supprimez des paquets installés avec un profil, tous les paquets dont le nom correspond aux paquets installés par le profil sont supprimés. Cela inclut leurs dépendances, à l'exception des paquets requis par un profil différent.

Pour supprimer les paquets d'un profil sélectionné, procédez comme suit.

Conditions préalables

- Le profil sélectionné a été installé à l'aide de la commande **dnf module install *module-name:stream/profile*** ou comme profil par défaut en utilisant la commande **dnf install *module-name:stream command***.
- Vous comprenez la [résolution modulaire des dépendances](#).

Procédure

1. Désinstaller les paquets appartenant au profil sélectionné :

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

Remplacez *module-name*, *stream*, et *profile* par le module, le flux et le profil que vous souhaitez désinstaller.

Il est également possible de désinstaller les paquets de tous les profils installés dans un flux :

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

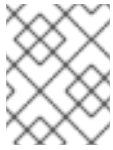
Ces opérations ne suppriment pas les paquets du flux qui n'appartiennent à aucun des profils.

2. Vérifiez la liste des paquets sous **Removing:** et **Removing unused dependencies:** avant de procéder à la suppression.

Pour retirer tous les paquets d'un flux sélectionné, suivez les instructions de la [section 8.3.2 "Retirer tous les paquets d'un flux de modules"](#).

Exemple 8.1. Suppression des paquets d'un profil sélectionné

L'exemple suivant montre comment supprimer les paquets et leurs dépendances qui appartiennent au profil **development** du flux de modules **nodejs:18**.



NOTE

Les résultats de cet exemple ont été édités pour des raisons de concision. Les sorties réelles peuvent contenir plus d'informations que celles présentées ici.

1. Installez le flux de modules **nodejs:18**, y compris tous les profils disponibles :

```
# dnf module install nodejs:18/*
(...)
Dependencies resolved.
=====
=====
Package      Architecture Version
Repository   Size
=====
=====
Installing group/module packages:
nodejs       x86_64      1:18.7.0-1.module+el9.1.0+16284+4fdefb2f
  rhel-AppStream 12 M
nodejs-devel x86_64      1:18.7.0-1.module+el9.1.0+16284+4fdefb2f
  rhel-AppStream 202 k
nodejs-nodemon noarch      2.0.15-1.module+el9.1.0+15718+e52ec601
  rhel-AppStream 762 k
npm          x86_64      1:8.15.0-1.18.7.0.1.module+el9.1.0+16284+4fdefb2f
  rhel-AppStream 2.2 M
Installing dependencies:
broccoli     x86_64      1.0.9-6.el9
  rhel-AppStream 314 k
broccoli-devel x86_64      1.0.9-6.el9
  rhel-AppStream 36 k
...
Installing weak dependencies:
nodejs-docs  noarch      1:18.7.0-1.module+el9.1.0+16284+4fdefb2f
  rhel-AppStream 7.2 M
nodejs-full-i18n x86_64      1:18.7.0-1.module+el9.1.0+16284+4fdefb2f
  rhel-AppStream 8.2 M
Installing module profiles:
nodejs/common
nodejs/development
nodejs/minimal
nodejs/s2i

Transaction Summary
=====
=====
Install 31 Packages

Total download size: 36 M
Installed size: 165 M
```

```
Is this ok [y/N]: y
(...)
Complete!
```

- Inspecter les profils installés :

```
$ dnf module info nodejs
...
Name       : nodejs
Stream     : 18 [e] [a]
Version    : 9010020221009220316
Context    : rhel9
Architecture : x86_64
Profiles   : common [d] [i], development [i], minimal [i], s2i [i]
Default profiles : common
Repo       : rhel-AppStream
...
Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled, [a]ctive
```

Tous les profils sont installés comme indiqué dans le résultat.

- Supprime les paquets du profil **development** et leurs dépendances :

```
# dnf module remove nodejs:18/development
(...)
Dependencies resolved.
=====

Package      Architecture Version
Repository    Size
=====

Removing:
nodejs-devel  x86_64      1:18.7.0-1.module+el9.1.0+16284+4fdefb2f
 @rhel-AppStream 950 k
Removing unused dependencies:
brotili       x86_64      1.0.9-6.el9
 @rhel-AppStream 754 k
brotili-devel x86_64      1.0.9-6.el9
 @rhel-AppStream 55 k
...
Disabling module profiles:
nodejs/development

Transaction Summary
=====

Remove 26 Packages

Freed space: 8.3 M
Is this ok [y/N]: y
```

- Inspecter les profils installés après le retrait :

```
$ dnf module info nodejs
```



```

...
Name       : nodejs
Stream     : 18 [e] [a]
Version    : 9010020221009220316
Context    : rhel9
Architecture : x86_64
Profiles   : common [d] [i], development, minimal [i], s2i [i]
Default profiles : common
Repo       : rhel-AppStream
Summary    : Javascript runtime
...
Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled, [a]ctive

```

Tous les profils, à l'exception de **development**, sont actuellement installés.

8.3.2. Suppression de tous les paquets d'un flux de modules

Lorsque vous supprimez des paquets installés avec un flux de modules, tous les paquets dont le nom correspond aux paquets installés par le flux sont supprimés. Cela inclut leurs dépendances, à l'exception des paquets requis par d'autres modules.

Pour supprimer tous les paquets d'un flux de modules sélectionné, procédez comme suit.

Conditions préalables

- Le flux de modules a été activé et au moins quelques paquets du flux ont été installés.
- Vous comprenez la [résolution modulaire des dépendances](#).

Procédure

1. Supprimer tous les paquets d'un flux sélectionné :

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

Remplacez *module-name* et *stream* par le module et le flux que vous souhaitez désinstaller.

2. Vérifiez la liste des paquets sous **Removing:** et **Removing unused dependencies:** avant de procéder à la suppression.
3. Optionnellement, réinitialiser ou désactiver le flux.

Si vous souhaitez supprimer uniquement les paquets d'un profil sélectionné, suivez les instructions de la [section 8.3.1. \ Supprimer les paquets d'un profil installé](#) .

Exemple 8.2. Suppression des paquets dans l'ensemble du flux

L'exemple suivant montre comment supprimer tous les paquets du flux de modules **nodejs:18**.



NOTE

Les résultats de cet exemple ont été édités pour des raisons de concision. Les sorties réelles peuvent contenir plus d'informations que celles présentées ici.

1. Installez le flux de modules **nodejs:18**, y compris tous les profils disponibles :

```
# dnf module install nodejs:18/*
(...)
Dependencies resolved.
=====
=====
Package      Architecture Version
Repository   Size
=====
=====
Installing group/module packages:
nodejs       x86_64      1:18.10.0-3.module+el9.1.0+16866+0fab0697
  rhel-AppStream 13 M
nodejs-devel x86_64      1:18.10.0-3.module+el9.1.0+16866+0fab0697
  rhel-AppStream 203 k
nodejs-nodemon noarch      2.0.15-1.module+el9.1.0+15718+e52ec601
  rhel-AppStream 762 k
npm          x86_64      1:8.19.2-1.18.10.0.3.module+el9.1.0+16866+0fab0697
  rhel-AppStream 2.2 M
Installing dependencies:
brotli       x86_64      1.0.9-6.el9
  rhel-AppStream 314 k
brotli-devel x86_64      1.0.9-6.el9
  rhel-AppStream 36 k
...
Installing weak dependencies:
nodejs-docs  noarch      1:18.10.0-3.module+el9.1.0+16866+0fab0697
  rhel-AppStream 7.3 M
nodejs-full-i18n x86_64      1:18.10.0-3.module+el9.1.0+16866+0fab0697
  rhel-AppStream 8.2 M
Installing module profiles:
nodejs/common
nodejs/development
nodejs/minimal
nodejs/s2i
Enabling module streams:
nodejs          18

Transaction Summary
=====
=====
Install 31 Packages

Total download size: 37 M
Installed size: 167 M
Is this ok [y/N]: y
```

2. Supprimer tous les paquets du flux de modules **nodejs:18**:

```
# dnf module remove --all nodejs:18
(...)
Dependencies resolved.
=====
=====
Package      Architecture Version
```

```

Repository      Size
=====
Removing:
nodejs          x86_64      1:18.10.0-3.module+el9.1.0+16866+0fab0697
  @rhel-AppStream 43 M
nodejs-devel    x86_64      1:18.10.0-3.module+el9.1.0+16866+0fab0697
  @rhel-AppStream 953 k
nodejs-docs     noarch      1:18.10.0-3.module+el9.1.0+16866+0fab0697
  @rhel-AppStream 78 M
nodejs-full-i18n x86_64      1:18.10.0-3.module+el9.1.0+16866+0fab0697
  @rhel-AppStream 29 M
nodejs-nodemon  noarch      2.0.15-1.module+el9.1.0+15718+e52ec601
  @rhel-AppStream 2.0 M
nodejs-packaging noarch      2021.06-4.module+el9.1.0+15718+e52ec601
  @rhel-AppStream 41 k
npm             x86_64      1:8.19.2-1.18.10.0.3.module+el9.1.0+16866+0fab0697
  @rhel-AppStream 6.9 M
Removing unused dependencies:
brotli         x86_64      1.0.9-6.el9
  @rhel-AppStream 754 k
brotli-devel   x86_64      1.0.9-6.el9
  @rhel-AppStream 55 k
...
Disabling module profiles:
nodejs/common
nodejs/development
nodejs/minimal
nodejs/s2i

Transaction Summary
=====
Remove 31 Packages

Freed space: 167 M
Is this ok [y/N]: y

```

3. Inspectez le module **nodejs** après l'avoir retiré :

```

$ dnf module info nodejs
...
Name       : nodejs
Stream     : 18 [e] [a]
Version    : 9010020221009220316
Context    : rhel9
Architecture : x86_64
Profiles   : common [d], development, minimal, s2i
Default profiles : common
...
Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled, [a]ctive

```

Aucun flux de module **Node.js** n'est actuellement installé.

Ressources supplémentaires

- [Réinitialisation des flux de modules](#)
- [Désactivation de tous les flux d'un module](#)

8.4. RESSOURCES SUPPLÉMENTAIRES

- [Commandes de suppression de contenu dans RHEL 9](#)

CHAPITRE 9. GESTION DE L'HISTORIQUE DE LA GESTION DES PAQUETS

La commande **dnf history** vous permet de consulter les informations suivantes :

- Calendrier des **DNF** transactions
- Dates et heures des transactions
- Nombre de paquets concernés par les transactions
- Si les transactions ont abouti ou ont été interrompues
- Si la base de données RPM a été modifiée entre les transactions

Vous pouvez également utiliser la commande **dnf history** pour annuler ou rétablir les transactions.

Dans les sections suivantes, vous apprendrez à gérer l'historique de la gestion des paquets à l'aide de la commande **dnf history**:

- [Liste des dernières transactions, des dernières opérations pour un paquet sélectionné et des détails d'une transaction particulière.](#)
- [Annulez les transactions **DNF**](#) en utilisant les commandes **dnf history undo** et **dnf history rollback**:
 - [Revenir sur une transaction unique **DNF**](#) .
 - [Annule toutes les transactions **DNF**](#) effectuées entre la transaction spécifiée et la dernière transaction.
- [Répéter les transactions sélectionnées ou les dernières transactions](#) .

9.1. TRANSACTIONS D'INSCRIPTION

Pour dresser la liste des dernières **DNF** les dernières opérations pour un paquet sélectionné et les détails d'une opération particulière, procédez comme suit.

Procédure

- Pour afficher une liste de toutes les dernières **DNF** utilisez la touche :

```
# dnf history
```

- Pour afficher une liste de toutes les dernières opérations pour un paquet sélectionné, utilisez :

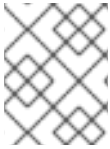
```
# dnf history list package-name
```

Remplacez *package-name* par le nom du paquet.

- Pour afficher les détails d'une transaction particulière, utilisez :

```
# dnf history info transactionID
```

Remplacez *transactionID* par l'identifiant de la transaction.



NOTE

Vous pouvez filtrer la sortie de la commande en ajoutant des expressions globales. Pour plus de détails, voir [Spécifier des expressions globales dans l'entrée `dnf`](#).

9.2. ANNULATION DES TRANSACTIONS DNF

Dans RHEL 9, vous pouvez annuler **DNF** de deux manières :

- Rétablir une transaction unique **DNF** la commande **`dnf history undo`** permet d'annuler une transaction unique.
- Annule toutes les transactions effectuées entre la transaction spécifiée et la dernière transaction en utilisant la commande **DNF** toutes les transactions effectuées entre la transaction spécifiée et la dernière transaction sont annulées à l'aide de la commande **`dnf history rollback`**.



IMPORTANT

Utilisez les commandes **`dnf history undo`** et **`dnf history rollback`** avec prudence. La rétrogradation des paquets RHEL, en particulier des paquets **`selinux`**, **`selinux-policy-*`**, **`kernel`**, **`glibc`** (dépendances de **`glibc`** telles que **`gcc`**), vers une version plus ancienne n'est pas prise en charge. Par conséquent, il n'est pas recommandé de rétrograder un système vers une version mineure (par exemple, de RHEL 9.1 à RHEL 9.0), car cela pourrait laisser le système dans un état indésirable.

9.2.1. L'annulation d'une transaction DNF unique en utilisant la fonction `dnf history undo`

Vous pouvez annuler les étapes effectuées au cours d'une seule transaction en utilisant la commande **`dnf history undo`**.

Si la transaction a installé un nouveau paquet, **`dnf history undo`** le désinstalle. Si la transaction a désinstallé un paquet, **`dnf history undo`** l'installe à nouveau. La commande **`dnf history undo`** tente également de rétrograder tous les paquets mis à jour à leur version précédente si celle-ci est encore disponible.



NOTE

Si une version plus ancienne du paquet n'est pas disponible, la rétrogradation à l'aide de la commande **`dnf history undo`** échoue.

Pour annuler une transaction unique à l'aide de la commande **`dnf history undo`**, procédez comme suit.

Procédure

1. Identifiez l'ID d'une transaction que vous souhaitez annuler :

```
# dnf history
```

2. Facultatif : Vérifiez qu'il s'agit bien de la transaction que vous souhaitez annuler en affichant ses détails :

```
# dnf history info transaction_id
```

Remplacez *transaction_id* par l'identifiant de la transaction que vous souhaitez annuler.

3. Annuler la transaction :

```
# dnf history undo transaction_id
```

Remplacez *transaction_id* par l'identifiant de la transaction que vous souhaitez annuler.

Exemple 9.1. Annulation d'une transaction DNF unique à l'aide de la commande `dnf history undo`

L'exemple suivant montre comment inverser la transaction d'installation du paquet **unzip** à l'aide de la commande **dnf history undo**.

1. Installez le paquetage **unzip**:

```
# dnf install unzip
Dependencies resolved.
=====
==
Package      Architecture  Version      Repository    Size
=====
==
Installing:
unzip        x86_64        6.0-56.el9   rhel          186 k

Transaction Summary
=====
==
Install 1 Package

Total download size: 186 k
Installed size: 392 k
Is this ok [y/N]: y
```

2. Identifiez l'ID de la transaction que vous souhaitez annuler :

```
# dnf history
ID | Command line | Date and time | Action(s) | Altered
-----
13 | install zip   | 2022-11-03 10:49 | Install   | 1
12 | install unzip | 2022-11-03 10:49 | Install   | 1
```

3. Annuler la transaction :

```
# dnf history undo 12
```

9.2.2. Annulation de plusieurs transactions DNF à l'aide de `dnf history rollback`

Vous pouvez annuler toutes les **DNF** effectuées entre une transaction spécifiée et la dernière transaction en utilisant la commande **dnf history rollback**. Notez que la transaction spécifiée par l'ID de la transaction restera inchangée.

Pour annuler plusieurs transactions à l'aide de la commande **dnf history rollback**, procédez comme suit.

Procédure

1. Identifiez l'identifiant de transaction de l'état auquel vous souhaitez revenir :

```
# dnf history
```

2. Rétablir les transactions spécifiées :

```
# dnf history rollback transaction_id
```

Remplacez *transaction_id* par l'identifiant de transaction de l'état auquel vous souhaitez revenir.

Pour annuler toutes les transactions de l'historique des transactions, il est également possible d'utiliser l'ID de la transaction 1:

```
# dnf history rollback 1
```

Exemple 9.2. Annulation de plusieurs transactions DNF à l'aide de la commande **dnf history rollback**

L'exemple suivant montre comment annuler deux opérations d'installation des paquets **unzip** et **zip** à l'aide de la commande **dnf history rollback**.

1. Installez le paquetage **unzip**:

```
# dnf install unzip
Dependencies resolved.
```

```
=====
```

```
Package      Architecture  Version      Repository    Size
=====
```

```
Installing:
```

```
unzip        x86_64        6.0-56.el9   rhel          186 k
```

```
Transaction Summary
```

```
=====
```

```
Install 1 Package
```

```
Total download size: 186 k
```

```
Installed size: 392 k
```

```
Is this ok [y/N]: y
```

2. Installez le paquetage **wget**:

```
# dnf install wget
Dependencies resolved.
```



```
=====
Package      Architecture  Version      Repository   Size
=====
```

```
Installing:
```

```
wget      x86_64      1.21.1-7.el9  rhel         794 k
```

```
Transaction Summary
=====
```

```
Install 1 Package
```

```
Total download size: 794 k
```

```
Installed size: 3.1 M
```

```
Is this ok [y/N]: y
```

- Identifiez l'identifiant de transaction de l'état auquel vous souhaitez revenir :

```
# dnf history
```

```
ID | Command line | Date and time | Action(s) | Altered
```

```
-----
14 | install wget | 2022-11-03 10:49 | Install | 1
13 | install unzip | 2022-11-03 10:49 | Install | 1
12 | install vim-X11 | 2022-11-03 10:20 | Install | 171 EE
```

- Reprendre les deux dernières transactions :

```
# dnf history rollback 12
```

Cette action désinstalle les paquets **wget** et **unzip** mais laisse le paquet **vim-X11** installé.

9.3. RÉPÉTITION DES OPÉRATIONS

Vous pouvez répéter les étapes d'une transaction en utilisant la commande **dnf history redo**.

Pour répéter une transaction sélectionnée ou la dernière **DNF** la dernière transaction, suivez les étapes suivantes.

Procédure

- Pour répéter une opération particulière, utilisez :

```
# dnf history redo transactionID
```

Remplacez *transactionID* par l'identifiant de la transaction.

- Pour répéter la dernière transaction, utilisez :

```
# dnf history redo last
```

CHAPITRE 10. GESTION DES DÉPÔTS DE LOGICIELS PERSONNALISÉS

Vous pouvez configurer un référentiel dans le fichier `/etc/dnf/dnf.conf` ou dans un fichier `.repo` dans le répertoire `/etc/yum.repos.d/`.

Les informations de configuration de **DNF** et des utilitaires associés sont stockées dans le fichier `/etc/dnf/dnf.conf`. Ce fichier contient la section `[main]` et peut contenir une ou plusieurs sections `[repository]`, qui vous permettent de définir des options spécifiques au référentiel. Les valeurs que vous définissez dans les sections `[repository]` du fichier `/etc/dnf/dnf.conf` remplacent les valeurs définies dans la section `[main]`.

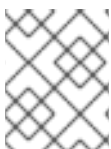
Toutefois, il est recommandé de définir des référentiels individuels dans les fichiers `.repo` nouveaux ou existants dans le répertoire `/etc/yum.repos.d/`.

Dans les sections suivantes, vous apprendrez à gérer des dépôts de logiciels personnalisés en utilisant la commande **DNF**:

- Définir les options de la section `[repository]` .
- Définir un nouveau référentiel **DNF** .
- Activer un dépôt **DNF** ajouté à votre système .
- Désactiver un dépôt **DNF** ajouté à votre système .

10.1. DÉFINITION DES OPTIONS DU RÉFÉRENTIEL DNF

Le fichier de configuration `/etc/dnf/dnf.conf` contient les sections `[repository]` où *repository* est un identifiant unique de dépôt. Les sections `[repository]` vous permettent de définir des référentiels individuels **DNF** référentiels individuels.



NOTE

Ne donnez pas aux dépôts personnalisés des noms utilisés par les dépôts Red Hat afin d'éviter les conflits.

Pour une liste complète des options disponibles `[repository]` voir la section `[repository] OPTIONS` de la page de manuel `dnf.conf(5)`.

10.2. AJOUTER UN RÉFÉRENTIEL DNF

Pour définir un nouveau référentiel, vous pouvez soit

- Ajouter une section `[repository]` au fichier `/etc/dnf/dnf.conf`.
- Ajouter une section `[repository]` à un fichier `.repo` dans le répertoire `/etc/yum.repos.d/`. Les RPM installés ou les outils de gestion de logiciels, par exemple le gestionnaire d'abonnement, peuvent fournir leur propre fichier `.repo`.



NOTE

Définissez vos dépôts dans un fichier **.repo** au lieu de **/etc/dnf/dnf.conf** car tous les fichiers avec l'extension **.repo** dans ce répertoire sont lus par **dnf**.

Pour ajouter un **DNF** à votre système en utilisant la commande **dnf config-manager**, suivez les étapes suivantes.

Procédure

- Ajouter un dépôt à votre système :

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

Remplacer *repository_url* par l'URL pointant vers le référentiel.



AVERTISSEMENT

L'obtention et l'installation de paquets logiciels à partir de sources non vérifiées ou non fiables autres que celles basées sur les certificats Red Hat **Content Delivery Network (CDN)** constituent un risque potentiel pour la sécurité et peuvent entraîner des problèmes de sécurité, de stabilité, de compatibilité et de maintenabilité.

10.3. ACTIVATION D'UN RÉFÉRENTIEL DNF

Pour activer un **DNF** ajouté à votre système à l'aide de la commande **dnf config-manager**, procédez comme suit.

Procédure

- Activer un référentiel :

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

Remplacer *repositoryID* par l'identifiant unique du référentiel.

Ressources supplémentaires

[Lister les référentiels](#)

10.4. DÉSACTIVATION D'UN RÉFÉRENTIEL DNF

Pour désactiver un **DNF** ajouté à votre système à l'aide de la commande **dnf config-manager**, procédez comme suit.

Procédure

- Désactiver un référentiel :

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

Remplacer *repositoryID* par l'identifiant unique du référentiel.

Ressources supplémentaires

[Lister les référentiels](#)

CHAPITRE 11. GESTION DES VERSIONS DU CONTENU DES FLUX D'APPLICATIONS

Le contenu du référentiel AppStream peut être disponible en plusieurs versions, correspondant à des flux de modules.

Dans les sections suivantes, vous apprendrez les opérations à effectuer lorsque vous modifiez des flux de modules activés existants :

- [Règles de dépendance modulaires](#) .
- [Interaction des dépendances modulaires et non modulaires](#) .
- [Réinitialiser les modules à leur état initial](#) .
- [Désactiver complètement un module et tous ses flux](#) .

11.1. DÉPENDANCES MODULAIRES ET CHANGEMENTS DE FLUX

Traditionnellement, les paquets fournissant du contenu dépendent d'autres paquets et spécifient généralement les versions de dépendance souhaitées. Pour les paquets contenus dans les modules, ce mécanisme s'applique également, mais le regroupement des paquets et de leurs versions particulières dans des modules et des flux fournit des contraintes supplémentaires. En outre, les flux de modules peuvent déclarer des dépendances sur les flux d'autres modules, indépendamment des paquets qu'ils contiennent et qu'ils fournissent.

Après toute opération avec des paquets ou des modules, l'arbre de dépendance complet de tous les paquets installés sous-jacents doit satisfaire à toutes les conditions déclarées par les paquets. En outre, toutes les dépendances de flux de modules doivent être satisfaites.

En conséquence :

- L'activation d'un flux de modules peut nécessiter l'activation d'autres flux de modules.
- L'installation d'un profil de flux de modules ou l'installation de paquets à partir d'un flux peut nécessiter l'activation d'autres flux de modules et l'installation d'autres paquets.
- La désactivation d'un flux de modules peut nécessiter la désactivation d'autres flux de modules. Aucun paquet ne sera supprimé automatiquement.
- La suppression d'un paquet peut nécessiter la suppression d'autres paquets. Si ces paquets étaient fournis par des modules, les flux de modules restent activés en vue d'une nouvelle installation, même si aucun paquet de ces flux n'est plus installé. Cela reflète le comportement d'un dépôt **DNF** inutilisé.

11.2. INTERACTION DES DÉPENDANCES MODULAIRES ET NON MODULAIRES

Les [dépendances modulaires](#) sont une couche supplémentaire qui s'ajoute aux dépendances RPM habituelles. Les dépendances modulaires se comportent de la même manière que les dépendances hypothétiques entre les dépôts. Cela signifie que l'installation de différents paquets nécessite la résolution à la fois des dépendances RPM et des dépendances modulaires.

Le système conservera toujours les choix de module et de flux, à moins qu'il ne soit explicitement demandé de les modifier. Un paquet modulaire recevra les mises à jour contenues dans le flux

actuellement activé du module qui fournit ce paquet, mais ne passera pas à une version contenue dans un flux différent.

11.3. RÉINITIALISATION DES FLUX DE MODULES

La réinitialisation d'un module est une action qui ramène tous ses flux à leur état initial – ni activés, ni désactivés. Si le module dispose d'un flux par défaut configuré, ce flux devient actif à la suite de la réinitialisation du module.

Pour rétablir l'état initial d'un flux de modules à l'aide de la commande **dnf module reset**, procédez comme suit.

Procédure

- Réinitialiser l'état du module :

```
# dnf module reset module-name
```

Le module est ramené à l'état initial. Les informations relatives à un flux activé et aux profils installés sont effacées, mais aucun contenu installé n'est supprimé.

11.4. DÉSACTIVATION DE TOUS LES FLUX D'UN MODULE

Les modules qui ont un flux par défaut auront toujours un flux actif. Dans les situations où le contenu de tous les flux du module ne doit pas être accessible, il est possible de désactiver l'ensemble du module.

Pour désactiver tous les flux d'un module à l'aide de la commande **dnf module disable**, procédez comme suit.

Conditions préalables

- Vous comprenez le [concept](#) de *active module stream*.

Procédure

- Désactiver le module :

```
# dnf module disable module-name
```

La commande **dnf** demande une confirmation, puis désactive le module et tous ses flux. Tous les flux du module deviennent inactifs. Aucun contenu installé n'est supprimé.

ANNEXE A. LISTE DES COMMANDES DNF

Dans les sections suivantes, vous examinerez **DNF** pour lister, installer et supprimer du contenu dans Red Hat Enterprise Linux 9.

A.1. COMMANDES POUR LISTER LE CONTENU DANS RHEL 9

Les commandes suivantes sont couramment utilisées pour **DNF** pour trouver du contenu et ses détails dans Red Hat Enterprise Linux 9 :

Commandement	Description
dnf search <i>term</i>	Recherchez un paquet en utilisant des termes liés à ce paquet.
dnf repoquery <i>package</i>	Recherche des dépôts DNF pour un paquet sélectionné et sa version.
dnf list	Liste des informations sur tous les paquets installés et disponibles.
dnf list --installed dnf repoquery --installed	Liste de tous les paquets installés sur votre système.
dnf list --available dnf repoquery	Liste de tous les paquets dans tous les dépôts activés qui sont disponibles pour l'installation.
dnf repolist	Liste de tous les dépôts activés sur votre système.
dnf repolist --disabled	Liste de tous les dépôts désactivés sur votre système.
dnf repolist --all	Liste des dépôts activés et désactivés.
dnf repoinfo	Liste d'informations supplémentaires sur les référentiels.
dnf info <i>package-name</i> dnf repoquery --info <i>package_name</i>	Affiche les détails d'un paquet disponible.
dnf repoquery --info --installed <i>package_name</i>	Affiche les détails d'un paquet installé sur votre système.
dnf module list	Liste des modules et de leur état actuel.
dnf module info <i>module-name</i>	Afficher les détails d'un module.

Commandement	Description
dnf module list <i>module-name</i>	Affiche l'état actuel d'un module.
dnf module info --profile <i>module-name</i>	Afficher les paquets associés aux profils disponibles d'un module sélectionné.
dnf module info --profile <i>module-name:stream</i>	Affiche les paquets associés aux profils disponibles d'un module en utilisant un flux spécifié.
dnf module provides <i>package</i>	Déterminer quels modules, flux et profils fournissent un paquet. Notez que si le paquet est disponible en dehors de tout module, la sortie de cette commande est vide.
dnf group summary	Afficher le nombre de groupes installés et disponibles.
dnf group list	Liste de tous les groupes installés et disponibles.
dnf group info <i>group-name</i>	Liste des paquets obligatoires et optionnels inclus dans un groupe particulier.

A.2. COMMANDES POUR L'INSTALLATION DE CONTENU DANS RHEL 9

Les commandes suivantes sont couramment utilisées pour l'installation de contenu dans Red Hat Enterprise Linux 9 : **DNF** pour installer du contenu dans Red Hat Enterprise Linux 9 :

Commandement	Description
dnf install <i>package-name</i>	Installer un paquet. Si le paquet est fourni par un flux de modules, dnf résout le flux de modules requis et l'active automatiquement lors de l'installation de ce paquet. Cela se produit également de manière récursive pour toutes les dépendances du paquet. Si d'autres flux de modules répondent à la demande, les flux par défaut sont utilisés.
dnf install <i>package-name-1 package-name-2</i>	Installer simultanément plusieurs paquets et leurs dépendances.
dnf install <i>package-name.arch</i>	Spécifiez l'architecture du paquet en l'ajoutant au nom du paquet lors de l'installation de paquets sur un système <i>multilib</i> (AMD64, Intel 64 machine).

Commandement	Description
dnf install /usr/sbin/binary-file	Installer un binaire en utilisant le chemin d'accès au binaire comme argument.
dnf install /path/	Installer un paquet précédemment téléchargé à partir d'un répertoire local.
dnf install package-url	Installer un paquetage distant à l'aide d'une URL de paquetage.
dnf module enable module-name:stream	Activer un module en utilisant un flux spécifique. Notez que l'exécution de cette commande n'installe aucun paquetage RPM.
dnf module install module-name:stream dnf install @module-name:stream	Installer un profil par défaut à partir d'un flux de modules spécifique. Notez que l'exécution de cette commande active également le flux spécifié.
dnf module install module-name:stream/profile dnf install @module-name:stream/profile	Installer un profil sélectionné en utilisant un flux spécifique.
dnf group install group-name	Installer un groupe de paquets par un nom de groupe.
dnf group install groupID	Installer un groupe de paquets en fonction de l'identifiant du groupe.

A.3. COMMANDES DE SUPPRESSION DE CONTENU DANS RHEL 9

Les commandes suivantes sont couramment utilisées pour supprimer du contenu dans Red Hat Enterprise Linux 9 : DNF couramment utilisées pour supprimer du contenu dans Red Hat Enterprise Linux 9 :

Commandement	Description
dnf remove package-name	Supprimer un paquet particulier et tous les paquets dépendants.
dnf remove package-name-1 package-name-2	Supprimer simultanément plusieurs paquets et leurs dépendances inutilisées.
dnf group remove group-name	Supprimer un groupe de paquets par le nom du groupe.

Commandement	Description
dnf group remove <i>groupID</i>	Supprime un groupe de paquets en fonction de l'identifiant du groupe.
dnf module remove --all <i>module-name:stream</i>	Supprime tous les paquets du flux spécifié. Notez que l'exécution de cette commande peut supprimer des paquets critiques de votre système.
dnf module remove <i>module-name:stream/profile</i>	Supprime les paquets d'un profil installé.
dnf module remove <i>module-name:stream</i>	Supprime les paquets de tous les profils installés dans le flux spécifié.
dnf module reset <i>module-name</i>	Réinitialiser un module à l'état initial. Notez que l'exécution de cette commande ne supprime pas les paquets du module spécifié.
dnf module disable <i>module-name</i>	Désactive un module et tous ses flux. Notez que l'exécution de cette commande ne supprime pas les paquets du module spécifié.