



## Cisco NX-OS ソフトウェアのオプション

この章では、Cisco NX-OS ソフトウェアのオプション性について説明します。

- [Cisco NX-OS ソフトウェアのオプション \(1 ページ\)](#)
- [モジュラ パッケージの使用 \(3 ページ\)](#)
- [Cisco NX-OS ソフトウェア パッケージのリスト \(4 ページ\)](#)
- [ベース モードまたはフル モードでの NX-OS イメージの起動 \(6 ページ\)](#)
- [ISSU のサポート \(6 ページ\)](#)
- [RPM に関する情報 \(6 ページ\)](#)
- [YUM コマンドに関する情報 \(23 ページ\)](#)
- [インストール操作ユーザ ロールの作成 \(42 ページ\)](#)
- [Compacting Cisco NX-OS ソフトウェア イメージの圧縮 \(42 ページ\)](#)

## Cisco NX-OS ソフトウェアのオプション

Cisco NX-OS リリース 9.2 (1) 以降では、Cisco NX-OS ソフトウェア イメージでモジュラ パッケージの管理をサポートします。これにより、Cisco NX-OS ソフトウェアは、基礎となる NX-OS ソフトウェアを変更することなく、機能を選択的に追加、削除、およびアップグレードする柔軟性を提供します。

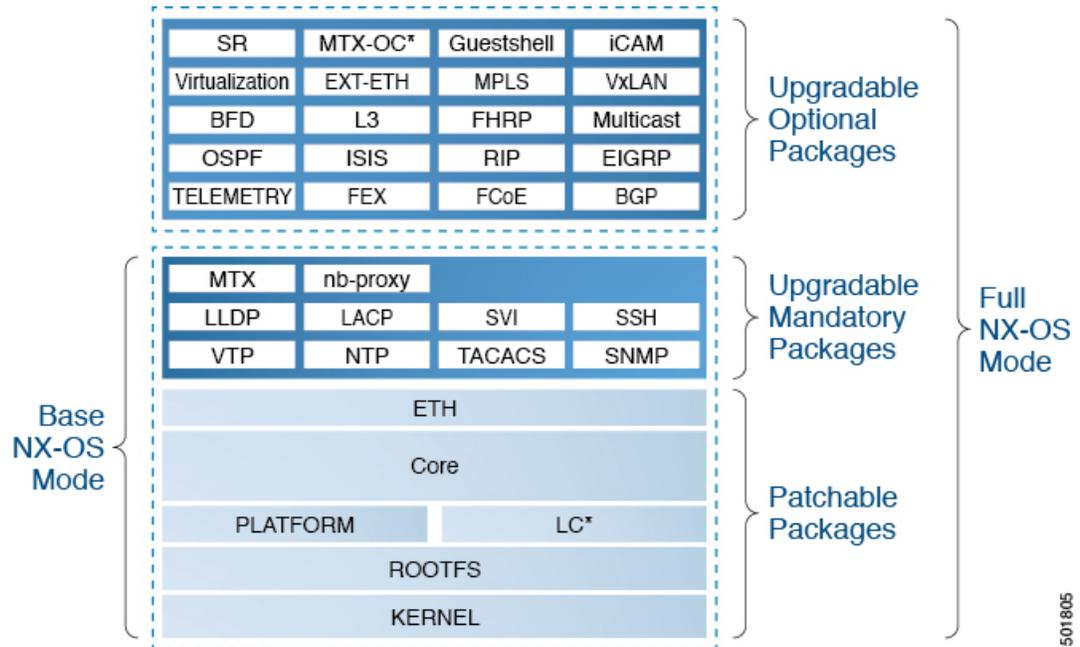
モジュール式の Cisco NX-OS ソフトウェアを使用する利点は次のとおりです。

- 無駄のない NX-OS ソフトウェア
- 機能と修正の非同期的な提供：クイックフィックス（新機能を含む）はリリースとは独立して提供されます。
- 実行時のバイナリとライブラリのフットプリントの削減

Cisco NX-OS ソフトウェアは、次の図に示すように2つのモードで NX-OS ソフトウェアをブートするようにプロビジョニングされています。

- ベース NX-OS モード
- フル NX-OS モード

図 1: Cisco NX-OS ソフトウェアのオプション



- ベース NX-OS モードには次が含まれます。
  - アップグレード可能な必須パッケージ
  - パッチ適用可能パッケージ
- フル NX-OS モードには次が含まれます。
  - アップグレード可能なオプションパッケージ
  - アップグレード可能な必須パッケージ
  - パッチ適用可能パッケージ



(注) デフォルトのモードは、フル NX-OS モードです。

ベース NX-OS モードでは、レイヤ 2 およびレイヤ 3 の基本的な機能が提供されます。すべてのダイナミックルーティング機能（BGP、OSPF、EIGRP、RIP、ISIS など）やその他のオプション機能 RPM はデフォルトでは使用できません。オプションの機能 RPM は、ベースイメージの上にインストールする必要があります。

フル NX-OS モードでは、ブート時にイーサネットプラグインがプラグインマネージャによりアクティブ化されるときにすべての機能 RPM がインストールされます。以前のリリースと比較して、ユーザの動作に変更はありません。

# モジュラ パッケージの使用

Cisco NX-OS ソフトウェア イメージは、以前から Cisco Linux ディストリビューションを形成するパッケージングで構成されています。各パッケージのサイズが大きいため、特定のパッケージのアップグレードが困難になっています。

このセクションでは、Cisco NX-OS ソフトウェア イメージの新しいパッケージの管理について説明します。Cisco NX-OS リリース 9.2(1)以降では、BGP、OSPF、VXLAN、MPLS、セグメントルーティングなどの一部の NX-OS 機能はオプションと見なされます。

各モジュラ パッケージには、次の重要な特徴があります。

- アップグレード機能：モジュラ パッケージは個別にアップグレード可能です。モジュラ パッケージは、同じリリースのものを使用する必要があります。複数のリリースにまたがるパッケージでのアップグレードの実行はサポートされていません。
- オプション性：モジュラ パッケージはオプションです。たとえば、これらのパッケージは実行時に削除またはアンインストールが可能です。モジュラ パッケージの削除はシステムの稼働に影響を与えず、スイッチのその他の機能にも影響を与えません。



- (注) モジュラ パッケージでエクスポートされたすべての API は、機能のインストール後のみ使用する必要があります。

## RPM と YUM

RPM (Red Hat Package Manager) は、Linux Standard Base (LSB) 内のパッケージングに使用されるパッケージ管理システムです。RPM コマンド オプションは、次の 3 つのサブグループにまとめられます。

- パッケージのクエリと確認
- パッケージのインストール、アップグレードおよび削除
- その他の機能の実行

**rpm** は RPM で使用されるメイン コマンドのコマンド名です。一方、**.rpm** は RPM ファイルに使用される拡張子です。

YUM (Yellowdog Updater, Modified) は、RPM ベース Linux システム用のオープン ソース コマンド ライン ツールです。これにより、ユーザとシステム管理者はシステム上のソフトウェア パッケージのインストール、アップデート、削除、または検索を簡単に行うことができます。YUM により、自動アップデートとパッケージ管理 (依存関係管理を含む) の機能がシステムに追加されます。YUM は、システムにインストールされたパッケージを把握するだけでなく、パッケージのコレクションであるリポジトリと連携します。通常、リポジトリにはネットワーク接続を介してアクセスできます。

## Cisco NX-OS ソフトウェア パッケージのリスト

Cisco NX-OS ソフトウェア イメージは、サードパーティ パッケージで構成されています。

- アップグレード可能なオプション パッケージ
- アップグレード可能な必須パッケージ
- パッチ適用可能パッケージ

### アップグレード可能なオプション パッケージ

アップグレード可能なオプション パッケージには、それぞれ次の重要な特徴があります。

- 個別にアップグレードが可能です。
- これらのパッケージはオプションであるため、実行時の削除やアンインストールなどが可能です。アップグレード可能なオプションパッケージの削除はシステムの稼働に影響を与えず、スイッチのその他の機能にも影響を与えません。



---

(注) オプションのパッケージによりエクスポートされたすべての API は、機能の存在を検出した後でのみ使用してください。

---

- これらのパッケージは、アップグレード、ダウングレード、アクティブ化、または非アクティブが可能です。

アップグレード可能なオプション パッケージには、次の項目の一部が含まれています。

- BGP
- BFD
- EIGRP
- Ext-Eth
- FCoE
- FEX
- FHRP
- Guestshell
- ISIS
- L3
- MPLS
- MTX-OC
- マルチキャスト

- OPENSSSH
- OSPF
- RIP
- SR
- TACACS+
- テレメトリ
- 仮想化
- VXLAN

#### アップグレード可能な必須パッケージ

必須のパッケージはアップグレードまたはダウングレードのみが可能であり、非アクティブ化はできません。パッケージで可能なステータスは、**active/install** または **inactive/uninstall** です。アップグレード可能な必須パッケージには、次のものが含まれています。

- LACP
- LLDP
- MTX
- nb-proxy
- NTP
- SNMP
- SSH
- SVI
- TACACS
- VTP

#### パッチ適用可能パッケージ

パッチ適用可能パッケージには、次のものが含まれています。

- カーネル
- LC
- Network-infra (別名 Eth)
- プラットフォーム
- Rootfs
- System-infra (別名 Core)

## ベース モードまたはフル モードでの NX-OS イメージの起動

NX-OS イメージは、ベース モードまたはフルモードでブートできるようになりました。フルブートモードでは、以前のリリースのソフトウェアと同様な完全な NX-OS ソフトウェアがインストールされます。これは、デフォルトのブートモードです。ベースブートモードでは、オプションの RPM はインストールされません。

コマンドライン オプションを使用するには、次の手順を参照してください。

- VSH プロンプトで **install reset nxos base** オプションを使用して、NX-OS イメージを基本ブートモードでインストールします。リロード後にスイッチはベースモードになり、オプションパッケージはインストールされません。
- VSH プロンプトで **install reset nxos full** オプションを使用して、NX-OS イメージをフルモードでインストールします。リロード後にスイッチはフルモードになり、オプションのパッケージが自動的にインストールされます。

詳細については、「機能 RPM の操作に install CLI を使用する」セクションを参照してください。

## ISSU のサポート

Cisco NX-OS リリース 9.2(1) 以降では、Cisco Nexus 9000 シリーズスイッチの ISSU に変更はありません。

## RPM に関する情報

RPM は、NX-OS の **install** コマンドを使用して、または **YUM** コマンドを使用して新しいソフトウェアバージョンにアップグレードまたはダウングレードできます。アップグレード可能な RPM には、オプションと必須があります。

オプションおよび必須の RPM の詳細については、以降のセクションを参照してください。

## オプション RPM とその関連機能

オプション RPM をインストールすると、ネイティブの NX-OS 動作に影響を与えずに機能を有効化できます。また、オプション RPM は、**install deactivate** コマンドを使用してスイッチから削除できます。

EIGRP などのオプション RPM は、基本ソフトウェアの一部ではありません。これらの RPM は、**yum** または **install CLI** コマンドを使用して、必要に応じてスイッチに対して追加、アップグレード、削除が可能です。

次のリストは、オプション RPM とその関連機能を示しています。

表 1: オプション RPM とその関連機能

| パッケージ名            | 関連機能  |
|-------------------|---|
| BGP               | feature bgp   |
| BFD               | feature bfd   |
| Container-tracker | feature container-tracker   |
| EIGRP             | feature eigrp   |
| Ext-Eth           | <ul style="list-style-type: none"> <li>• feature openflow</li> <li>• feature evb</li> <li>• feature imp</li> <li>• feature netflow</li> <li>• feature sla_sender</li> <li>• feature sla_responder</li> <li>• feature sla twamp-server</li> <li>• feature sflow</li> </ul> |
| FCoE              | <ul style="list-style-type: none"> <li>• feature-set fcoe</li> <li>• feature-set fcoe-npv</li> </ul>  |
| FEX               | feature-set fex   |
| FHRP              | <ul style="list-style-type: none"> <li>• feature hsrp</li> <li>• feature vrrpv3</li> </ul>  |
| iCAM              | feature icam  |
| ISIS              | feature isis  |
| MPLS              | <ul style="list-style-type: none"> <li>• feature mpls segment-routing</li> <li>• feature mpls evpn</li> </ul>   |
| マルチキャスト           | <ul style="list-style-type: none"> <li>• feature pim</li> <li>• feature pim6</li> <li>• feature msdp</li> <li>• feature ngmvpn</li> </ul>   |

| パッケージ名    | 関連機能  |
|-----------|---|
| OSPF      | <ul style="list-style-type: none"> <li>• feature ospf</li> <li>• feature ospfv3</li> </ul>                  |
| RIP       | feature rip   |
| サービス      | feature catena  |
| SR        | feature mpls segment-routing traffic-engineering  |
| TELEMETRY | feature telemetry   |
| 仮想化       | 該当なし  |
| VXLAN     | <ul style="list-style-type: none"> <li>• feature nv overlay</li> <li>• feature fabric forwarding</li> </ul> |

## NX-OS 機能 RPM インストールに関するガイドライン

Cisco NX-OS シリーズ スイッチに存在する次の RPM 管理用 NX-OS システム RPM リポジトリを参照してください。



- (注) RPM をシステムのリポジトリに手動でコピーをすることは避けてください。代わりに、install または YUM コマンドを使用してください。

表 2: スイッチに存在する RPM リポジトリ

| リポジトリ名      | リポジトリのパス | 説明  |
|-------------|----------|---|
| groups-repo | /rpms    | バンドルされている NX-OS イメージの一部です。NX-OS イメージの一部としてバンドルされているすべての RPM を保持するために使用されます。このリポジトリに格納されているすべての RPM は、基本 RPM と呼ばれます。 |

| リポジトリ名      | リポジトリのパス                                | 説明   |
|-------------|---|--|
| localdb     | /bootflash/.rpmstore/patching/localrepo | <p>RPM の保持に使用されます。ユーザが NX-OS 機能 RPM を <b>install add</b> コマンドの一部として追加すると、RPM がこの場所にコピーされ、リロード時に常駐します。ユーザは、リポジトリをクリーンアップする責任があります。</p> <p>このリポジトリに RPM を追加するには、<b>install add</b> コマンドを使用します。</p> <p>このリポジトリから RPM を削除するには、<b>install remove</b> コマンドを使用します。</p> <p>YUM コマンドを使用しても、リポジトリに追加できます。</p> <p>Cisco Nexus 3000 シリーズ スイッチを除き、リポジトリの最大領域は 200 Mb です (Cisco Nexus 9000 シリーズ スイッチのパッチ リポジトリを含む)。Cisco Nexus 3000 シリーズ スイッチでは、リポジトリの最大サイズは 20 Mb です。</p> |
| patching    | /bootflash/.rpmstore/patching/patchrepo | <p>RPM の保持に使用されます。ユーザが NX-OS パッチ RPM をスイッチに追加すると、パッチ RPM がこのリポジトリにコピーされます。</p>   |
| Third_Party | /bootflash/.rpmstore/thirdparty         | <p>ユーザがサードパーティ RPM を追加したときに、RPM の保持に使用されます。</p>  |

**groups-repo** と **localdb** リポジトリには、システム ブート時またはアクティベーション時にインストールする必要がある NX-OS 機能 RPM が保持されます。YUM コマンドまたは **install** コマンドは、これらの RPM のインストールまたは削除に使用できます。

次のルールが、ブートまたはインストール時の機能 RPM のインストール手順に適用されます。

- 同じ NX-OS リリース番号の RPM のみをインストール用を選択する必要があります。
- ベース RPM は、**localdb** リポジトリに追加できません。

## パッチ適用可能な NX-OS 必須 RPM のリスト

次に、パッチ適用可能な NX-OS 必須 RPM のリストを示します。

表 3: パッチ適用可能な NX-OS 必須 RPM のリスト

| Serial Number | RPM 名    | 説明  |
|---------------|----------|---|
| 1             | コア       | NX-OS インフラストラクチャソフトウェア。   |
| 2             | プラットフォーム | Cisco NX-OS プラットフォーム固有のソフトウェアおよび一部の Linux 変更済みソフトウェア。   |
| 3             | イーサネット   | インフラストラクチャと緊密に連携する Cisco NX-OS 機能。  |
| 4             | ラインカード   | Cisco Nexus 9000 プラットフォームスイッチ内の Cisco NX-OS arm ベースラインカード、および-Rシリーズラインカードを装備した Cisco Nexus 9508 プラットフォームスイッチ内の x86_64 ラインカード。 |
| 5             | ラインカード 2 | Cisco Nexus 9000 シリーズスイッチ内の Cisco NX-OS x85_64 ラインカード。  |
| 6             | TOR      | Cisco NX-OS トップオブラックスイッチ。   |

## 機能 RPM の操作に install CLI を使用する

機能 RPM の操作に install CLI を使用するには、次のリファレンス表を参照してください。

表 4: 機能 RPM 操作の **install CLI** リファレンス

| CLI                            | 説明  |
|--------------------------------|---|
| <b>install reset</b>           | <p>この操作は、すべてのパッチ、保持されたコンフィグレーション、アップグレードされたパッケージ、サードパーティのインストール済みパッケージ、未保存のコンフィグレーションを削除し、デフォルトのパッケージを使用してスイッチの以前のモード（フル/基本）をリロードします。</p> <p><b>install reset</b> コマンドも <b>write erase</b> 操作を実行します。次のメッセージがプロンプトに表示されます。</p> <pre>switch(config)# install reset</pre> <hr/> <p>WARNING!!This operation will remove all patches, upgraded packages, persisted etc configs, third party packages installed, startup configuration(write erase) and reload the switch with default packages.</p> <hr/> <p>Do you want to proceed with reset operation? (y/n)? [n]</p> |
| <b>install reset nxos base</b> | <p>この操作は、すべてのパッチ、アップグレードされたパッケージ、保持された etc コンフィグレーション、サードパーティのインストール済みパッケージ、スタートアップ設定（<b>write erase</b>）を削除して NX-OS をベース モードでインストールして、デフォルトのパッケージを使用してスイッチをリロードします。</p>  |
| <b>install reset nxos full</b> | <p>この操作は、すべてのパッチ、アップグレードされたパッケージ、保持された etc コンフィグレーション、サードパーティのインストール済みパッケージ、スタートアップ設定（<b>write erase</b>）を削除して NXOS をフルモードでインストールして、デフォルトのパッケージ（必須およびオプションの RPM による）を使用してスイッチをリロードします。</p>  |
| <b>install add &lt;&gt;</b>    | <p>それぞれのリポジトリに RPM ファイルを追加して、リポジトリ（<b>patch/feature/third-party</b>）を更新します。</p>   |

| CLI                                  | 説明   |
|--------------------------------------|--|
| <b>install activate</b> <rpm name>   | リポジトリに存在する RPM をインストールします。   |
| <b>install commit</b> <rpm name>     | パッチ RPM に使用します。リロード時にパッチを保持します。                                      |
| <b>install deactivate</b> <rpm name> | RPM をアンインストールします。  |
| <b>install remove</b> <rpm name>     | リポジトリから RPM ファイルを削除してリポジトリを更新します。                                    |
| <b>sh install active</b>             | ベース rootfs RPM 以外の、システムにインストールされている RPM のリストを表示します (機能/パッチ/サードパーティ)。 |
| <b>sh install inactive</b>           | リポジトリに保持されていてインストールされていない RPM のリストを表示します。                            |
| <b>sh install packages</b>           | Rootfs RPM を含む、インストールされているすべての RPM をリストします。                          |

## デジタル署名のサポートにインストール CLI を使用する

デジタル署名のサポートに install CLI を使用方法の詳細については、次のセクションを参照してください。

### 手順

#### ステップ 1 switch#install add bootflash:<keyfile> gpg-key

例 :

```
install add bootflash:RPM-GPG-KEY-puppetlabs gpg-key
[#####] 100%
Install operation 304 completed successfully at Thu Apr 19 16:40:28 2018
```

Cisco GPG (NU Privacy Guard) キーを使用して Cisco リリース RPM に署名します。公開 GPG キーは、`/etc/pki/rpm-gpg/arm-Nexus9k-rel.gpg` に存在します。異なるソースからその他の公開キーを追加するには、このセクションの手順を使用してください。

#### ステップ 2 switch#install verify package <package-name> OR switch#install verify bootflash:<RPM file>

例 :

```
switch# install verify bootflash:vxlan-2.0.0.0-9.2.1.lib32_n9000.rpm

RSA signed
```

```
switch#
```

RPM ファイルが署名済みか未署名であるかどうかを確認するために、CLI を表示します。

### ステップ 3 show install packages

署名済みまたは未署名の情報を含むすべてのパッケージを表示します。

例：

```
switch# sh install packages

Boot Image:
NXOS Image: bootflash:/nxos.9.2.1.bin

-----
Installed Packages
attr.x86_64 2.4.47-r0.0 installed Unsigned
aufs-util.x86_64 3.14+git0+b59a2167a1-r0.0 installed Unsigned
base-files.n9000 3.0.14-r89.0 installed Unsigned
base-passwd.lib32_x86 3.5.29-r0.1.0 installed Unsigned
bash.lib32_x86 4.3.30-r0.0 installed Unsigned
bfd.lib32_n9000 2.0.0.0-9.2.1 installed Signed
bgp.lib32_n9000 2.0.0.0-9.2.1 installed Signed
binutils.x86_64 2.25.1-r0.0 installed Unsigned
bridge-utils.x86_64 1.5-r0.0 installed Unsigned
busybox.x86_64 1.23.2-r0.0 installed Unsigned
busybox-udhcpc.x86_64 1.23.2-r0.0 installed Unsigned
bzip2.x86_64 1.0.6-r5.0 installed Unsigned
ca-certificates.all 20150426-r0.0 installed Unsigned
cgroup-lite.x86_64 1.1-r0.0 installed Unsigned
chkconfig.x86_64 1.3.58-r7.0 installed Unsigned
container-tracker.lib32_n9000 2.0.0.0-9.2.1 installed Signed
containerd-docker.x86_64 0.2.3+gitaa8187dbd3b7ad67d8e5e3a15115d3eef43a7ed1-r0.0
installed Unsigned
core.lib32_n9000 2.0.0.0-9.2.1 installed Signed
coreutils.lib32_x86 8.24-r0.0 installed Unsigned
cpio.x86_64 2.12-r0.0 installed Unsigned
cracklib.lib32_x86 2.9.5-r0.0 installed Unsigned
cracklib.x86_64 2.9.5-r0.0 installed Unsigned
createrepo.x86_64 0.4.11-r9.0 installed Unsigned
cronie.x86_64 1.5.0-r0.0 installed Unsigned
curl.lib32_x86 7.60.0-r0.0 installed Unsigned
db.x86_64 6.0.30-r0.0 installed Unsigned
dbus-1.lib32_x86 1.8.20-r0.0 installed Unsigned
dhcp-client.x86_64 4.3.2-r0.0 installed Unsigned
dhcp-server.x86_64 4.3.2-r0.0 installed Unsigned
switch#
```

## インストールされているすべての RPM のクエリ

インストール済みのすべての RPM をクエリするには、次のステップを実行します。

## 手順

|        | コマンドまたはアクション  | 目的                           |
|--------|---|------------------------------|
| ステップ 1 | <p><b>show install packages</b></p> <p>例 :</p> <pre>switch# show install packages  Boot Image: NXOS Image: bootflash:/nxos.9.2.1.bin</pre> <hr/> <pre>Installed Packages attr.x86_64 2.4.47-r0.0 installed Unsigned aufs-util.x86_64 3.14+git0+b59a2167a1-r0.0 installed Unsigned base-files.n9000 3.0.14-r89.0 installed Unsigned base-passwd.lib32_x86 3.5.29-r0.1.0 installed Unsigned bash.lib32_x86 4.3.30-r0.0 installed Unsigned bfd.lib32_n9000 2.0.0.0-9.2.1 installed Signed bgp.lib32_n9000 2.0.0.0-9.2.1 installed Signed binutils.x86_64 2.25.1-r0.0 installed Unsigned bridge-utils.x86_64 1.5-r0.0 installed Unsigned busybox.x86_64 1.23.2-r0.0 installed Unsigned busybox-udhcpc.x86_64 1.23.2-r0.0 installed Unsigned bzip2.x86_64 1.0.6-r5.0 installed Unsigned ca-certificates.all 20150426-r0.0 installed Unsigned cgroup-lite.x86_64 1.1-r0.0 installed Unsigned chkconfig.x86_64 1.3.58-r7.0 installed Unsigned container-tracker.lib32_n9000 2.0.0.0-9.2.1 installed Signed containerd-docker.x86_64 0.2.3+gitaa8187cd3b7ad67d8e5e3a15115d3eef43a7ed1-r0.0 installed Unsigned core.lib32_n9000 2.0.0.0-9.2.1 installed Signed coreutils.lib32_x86 8.24-r0.0 installed Unsigned cpio.x86_64 2.12-r0.0 installed Unsigned cracklib.lib32_x86 2.9.5-r0.0 installed Unsigned cracklib.x86_64 2.9.5-r0.0 installed Unsigned createrepo.x86_64 0.4.11-r9.0 installed Unsigned</pre> | インストールされているすべての RPM をクエリします。 |

|  | コマンドまたはアクション  | 目的 |
|--|---|----|
|  | <pre>cronie.x86_64 1.5.0-r0.0 installed Unsigned curl.lib32_x86 7.60.0-r0.0 installed Unsigned db.x86_64 6.0.30-r0.0 installed Unsigned dbus-1.lib32_x86 1.8.20-r0.0 installed Unsigned dhcp-client.x86_64 4.3.2-r0.0 installed Unsigned dhcp-server.x86_64 4.3.2-r0.0 installed Unsigned switch#</pre> |    |

## インストール済み 機能 NX-OS RPM のみのクエリ

インストール済みの NX-OS 機能 RPM のみをクエリするには、次のステップを実行します。

### 手順

|        | コマンドまたはアクション   | 目的                                |
|--------|--|-----------------------------------|
| ステップ 1 | <pre><b>show install packages   grep</b> &lt;lib32_n9000&gt;  例 :</pre> <pre>switch# <b>show install packages  </b> <b>grep lib32_n9000</b> core.lib32_n9000 2.0.0.0-9.2.1 installed Signed eth.lib32_n9000 2.0.0.0-9.2.1 installed Signed lACP.lib32_n9000 2.0.0.0-9.2.1 installed Signed linecard2.lib32_n9000 2.0.0.0-9.2.1 installed Signed lldp.lib32_n9000 2.0.0.0-9.2.1 installed Signed mtx-device.lib32_n9000 2.0.0.0-9.2.1 installed Signed mtx-grpc-agent.lib32_n9000 2.0.0.0-9.2.1 installed Signed mtx-infra.lib32_n9000 2.0.0.0-9.2.1 installed Signed mtx-netconf-agent.lib32_n9000 2.0.0.0-9.2.1 installed Signed mtx-restconf-agent.lib32_n9000</pre> | インストール済みの NX-OS 機能 RPM のみをクエリします。 |

## インストール済みサードパーティ RPM のみのクエリ

|  | コマンドまたはアクション   | 目的 |
|--|--|----|
|  | <pre> 2.0.0.0-9.2.1 installed          Signed mtx-telemetry.lib32_n9000 2.0.0.0-9.2.1 installed          Signed ntp.lib32_n9000 2.0.0.0-9.2.1 installed          Signed nxos-ssh.lib32_n9000 2.0.0.0-9.2.1 installed          Signed platform.lib32_n9000 2.0.0.0-9.2.1 installed          Signed snmp.lib32_n9000 2.0.0.0-9.2.1 installed          Signed svi.lib32_n9000 2.0.0.0-9.2.1 installed          Signed tacacs.lib32_n9000 2.0.0.0-9.2.1 installed          Signed tor.lib32_n9000 2.0.0.0-9.2.1 installed          Signed vtp.lib32_n9000 2.0.0.0-9.2.1 installed          Signed </pre> |    |

## インストール済みサードパーティ RPM のみのクエリ

インストール済みのサードパーティ RPM のみをクエリするには、次のステップを実行します。

## 手順

|        | コマンドまたはアクション   | 目的   |
|--------|--|--|
| ステップ 1 | <pre> <b>show install packages   grep &lt;x86_64&gt;</b> 例 :  switch# <b>show install packages  </b> <b>grep x86_64</b> attr.x86_64 2.4.47-r0.0 installed          Unsigned aufs-util.x86_64 3.14+git0+b59a2167a1-r0.0 installed          Unsigned base-passwd.x86_64 3.5.29-r0.1.0 installed          Unsigned binutils.x86_64 2.25.1-r0.0 installed          Unsigned </pre> | <p>インストール済みのサードパーティ RPM をクエリします。</p> <p>インストールされているすべての RPM をクエリします。</p> |

|  | コマンドまたはアクション  | 目的 |
|--|---|----|
|  | <pre>bridge-utils.x86_64 1.5-r0.0 installed           Unsigned busybox.x86_64 1.23.2-r0.0 installed           Unsigned busybox-udhcpc.x86_64 1.23.2-r0.0 installed           Unsigned bzip2.x86_64 1.0.6-r5.0 installed           Unsigned cgroup-lite.x86_64 1.1-r0.0 installed           Unsigned</pre> |    |

## 1 ステップ手順による RPM のインストール

RPM をインストールする CIL とアップグレードする CLI は同じです。1 ステップ手順を使用して RPM をインストールするには、次の手順を参照してください。

### 手順

|        | コマンドまたはアクション   | 目的                      |
|--------|--|-------------------------|
| ステップ 1 | <pre><b>install add &lt;rpm&gt;activate</b></pre> <p>例 :</p> <pre>switch# <b>install add</b> bootflash:chef.rpm <b>activate</b> Adding the patch (/chef.rpm) [#####] 100% Install operation 868 completed successfully at Tue May 8 11:20:10 2018  Activating the patch (/chef.rpm) [#####] 100% Install operation 869 completed successfully at Tue May 8 11:20:20 2018</pre> | RPM をインストールしてアクティブ化します。 |

### 例

```
switch# show install active
Boot Image:
    NXOS Image: bootflash:/nxos.9.2.1.bin

Active Packages:
bgp-2.0.1.0-9.2.1.lib32_n9000
chef-12.0.0alpha.2+20150319234423.git.1608.b6eb10f-1.e15.x86_64
```

```

Active Base Packages:
  lacp-2.0.0.0-9.2.1.lib32_n9000
  lldp-2.0.0.0-9.2.1.lib32_n9000
  mtx-device-2.0.0.0-9.2.1.lib32_n9000
  mtx-grpc-agent-2.0.0.0-9.2.1.lib32_n9000
  mtx-infra-2.0.0.0-9.2.1.lib32_n9000
  mtx-netconf-agent-2.0.0.0-9.2.1.lib32_n9000
  mtx-restconf-agent-2.0.0.0-9.2.1.lib32_n9000
  mtx-telemetry-2.0.0.0-9.2.1.lib32_n9000
  ntp-2.0.0.0-9.2.1.lib32_n9000
  nxos-ssh-2.0.0.0-9.2.1.lib32_n9000
  snmp-2.0.0.0-9.2.1.lib32_n9000
  svi-2.0.0.0-9.2.1.lib32_n9000
  tacacs-2.0.0.0-9.2.1.lib32_n9000
  vtp-2.0.0.0-9.2.1.lib32_n9000
switch(config)#

```

## 2 ステップ手順による RPM のインストール

RPM をインストールする CIL とアップグレードする CLI は同じです。2 ステップ手順を使用して RPM をインストールするには、以下の手順を参照してください。

### 手順

|        | コマンドまたはアクション   | 目的              |
|--------|--|-----------------|
| ステップ 1 | <pre>install add &lt;rpm&gt;</pre> <p>例 :</p> <pre>switch# install add bootflash:vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm</pre> <pre>[#####] 100% Install operation 892 completed successfully at Thu Jun  7 13:56:38 2018</pre> <pre>switch(config)# sh install inactive   grep vxlan</pre> <pre>vxlan-2.0.1.0-9.2.1.lib32_n9000</pre> | RPM をインストールします。 |
| ステップ 2 | <pre>install activate &lt;rpm&gt;</pre> <p>例 :</p>   | RPM をアクティブ化します。 |

### 例

```
switch#install activate vxlan
```

```
[#####] 100%
Install operation 891 completed successfully at Thu Jun  7 13:53:07 2018

switch# show install active | grep vxlan

vxlan-2.0.0.0-9.2.1.lib32_n9000

switch# sh install inactive | grep vxlan

switch#
```

## 1 ステップの RPM のアップグレード

RPM をインストールする CIL とアップグレードする CLI は同じです。RPM をアップグレードするには、以下の手順を参照してください。

### 手順

|        | コマンドまたはアクション  | 目的              |
|--------|---|-----------------|
| ステップ 1 | <b>install add &lt;rpm&gt;activate upgrade</b><br>例 :<br><br><pre>switch(config)# install add bootflash:bp-2.0.2.0-9.2.1.lib32_n9000.rpm activate upgrade</pre> <p>Adding the patch<br/>           (/bgp-2.0.2.0-9.2.1.lib32_n9000.rpm)<br/>           [#####] 100%<br/>           Install operation 870 completed<br/>           successfully at Tue May 8 11:22:30 2018</p> <p>Activating the patch<br/>           (/bgp-2.0.2.0-9.2.1.lib32_n9000.rpm)<br/>           [#####] 100%<br/>           Install operation 871 completed<br/>           successfully at Tue May 8 11:22:40 2018</p> | RPM をインストールします。 |

### 例

```
switch(config)# show install active

Boot Image:
  NXOS Image: bootflash:/nxos.9.2.1.bin

Active Packages:
  bgp-2.0.2.0-9.2.1.lib32_n9000
```

```
chef-12.0.0alpha.2+20150319234423.git.1608.b6eb10f-1.el5.x86_64
```

```
Active Base Packages:
  lacp-2.0.0.0-9.2.1.lib32_n9000
  lldp-2.0.0.0-9.2.1.lib32_n9000
  mtx-device-2.0.0.0-9.2.1.lib32_n9000
  mtx-grpc-agent-2.0.0.0-9.2.1.lib32_n9000
  mtx-infra-2.0.0.0-9.2.1.lib32_n9000
  mtx-netconf-agent-2.0.0.0-9.2.1.lib32_n9000
  mtx-restconf-agent-2.0.0.0-9.2.1.lib32_n9000
  mtx-telemetry-2.0.0.0-9.2.1.lib32_n9000
  ntp-2.0.0.0-9.2.1.lib32_n9000
  nxos-ssh-2.0.0.0-9.2.1.lib32_n9000
  snmp-2.0.0.0-9.2.1.lib32_n9000
  svi-2.0.0.0-9.2.1.lib32_n9000
  tacacs-2.0.0.0-9.2.1.lib32_n9000
  vtp-2.0.0.0-9.2.1.lib32_n9000
```

## RPM のダウングレード

ダウングレード手順では、特別な CLI 属性を必要とします。1 ステップ手順を使用して RPM をダウングレードするには、次の手順を参照してください。

### 手順

|        | コマンドまたはアクション   | 目的               |
|--------|--|------------------|
| ステップ 1 | <pre>install add &lt;rpm&gt;activate downgrade</pre> <p>例 :</p> <pre>switch(config)# install add bootflash:bp-2.0.1.0-9.2.1.lib32_n9000.rpm activate downgrade</pre> <pre>Adding the patch (/bgp-2.0.1.0-9.2.1.lib32_n9000.rpm) [#####] 100% Install operation 872 completed successfully at Tue May 8 11:24:43 2018</pre> <pre>Activating the patch (/bgp-2.0.1.0-9.2.1.lib32_n9000.rpm) [#####] 100% Install operation 873 completed successfully at Tue May 8 11:24:52 2018</pre> | RPM をダウングレードします。 |

### 例

```
switch(config)# show install active
Boot Image:
```

```

NXOS Image: bootflash:/nxos.9.2.1.bin

Active Packages:
  bgp-2.0.1.0-9.2.1.lib32_n9000
  chef-12.0.0alpha.2+20150319234423.git.1608.b6eb10f-1.e15.x86_64

Active Base Packages:
  lacp-2.0.0.0-9.2.1.lib32_n9000
  lldp-2.0.0.0-9.2.1.lib32_n9000
  mtx-device-2.0.0.0-9.2.1.lib32_n9000
  mtx-grpc-agent-2.0.0.0-9.2.1.lib32_n9000
  mtx-infra-2.0.0.0-9.2.1.lib32_n9000
  mtx-netconf-agent-2.0.0.0-9.2.1.lib32_n9000
  mtx-restconf-agent-2.0.0.0-9.2.1.lib32_n9000
  mtx-telemetry-2.0.0.0-9.2.1.lib32_n9000
  ntp-2.0.0.0-9.2.1.lib32_n9000
  nxos-ssh-2.0.0.0-9.2.1.lib32_n9000
  snmp-2.0.0.0-9.2.1.lib32_n9000
  svi-2.0.0.0-9.2.1.lib32_n9000
  tacacs-2.0.0.0-9.2.1.lib32_n9000
  vtp-2.0.0.0-9.2.1.lib32_n9000
switch(config)#

```

## RPM の削除

RPM を削除するには、次の手順を参照してください。

### 手順

|        | コマンドまたはアクション   | 目的                  |
|--------|--|---------------------|
| ステップ 1 | <pre> <b>install remove &lt;rpm&gt;</b>  例 :  switch(config)# <b>show install</b> <b>inactive   grep vxlan</b>            vxlan-2.0.0.0-9.2.1.lib32_n9000 switch(config)# <b>install remove</b> <b>vxlan</b>  Proceed with removing vxlan? (y/n)? [n] y [#####] 100% Install operation 890 Removal of base rpm package is not permitted at Thu Jun  7 13:52:15 2018 </pre> | リポジトリから、RPM を削除します。 |

## RPM の形式

RPM の一般的な形式は、<name>-<version>-<release>.<arch>.rpm です。同じ形式が NX-OS 機能 RPM にも適用されます。

- name : パッケージ名 (例 : BGP)
- version (<X.y.x.b> 形式) : <major.minor.patch.build\_number> (例 : 2.0.1.0)
- release : RPM 作成元のブランチ (例 : 9.2.1)
- arch : RPM のアーキテクチャタイプ (例 : lib32\_n9000)

次の表では、fex-2.0.0.0-9.2.1.lib32\_n9000.rpm を例として命名規則の詳細を説明しています。

表 5: RPM 命名規則

| RPM 命名規則                              | 説明  |
|---------------------------------------|---|
| 例 : fex-2.0.0.0-9.2.1.lib32_n9000.rpm |   |
| fex                                   | コンポーネントの名前を示しています。  |
| 2                                     | RPM に後方互換性がないことを示します。アップグレード中に設定の損失が発生します。  |
| 0                                     | 後方互換性がある増分 API 変更/CLI 変更/スキーマ変更を示します。既存の機能上の新しい機能が該当します。アップグレード中に失われる設定はありません。  |
| 0                                     | 機能の変更がないバグ修正を示します。アップグレード中に失われる設定はありません。  |
| 0                                     | この番号は、リリースの開発サイクルの間にコンポーネントが変更された回数を追跡します。この値はすべてのリリース イメージで 0 となります。   |
| 9.2.1                                 | RPM のリリース番号またはディストリビューションバージョンを示します。NVR 形式に沿っています。機能 RPM は NX-OS リリースにのみ適用可能であるため、このフィールドには存在する NX-OS リリースバージョンのみが指定されます。 |
| lib32_n9000                           | RPM のアーキテクチャタイプを示します。   |

## インストール中に RPM バージョンを管理するためのルール

**groups-repo** と **localdb** リポジトリには、システム ブート時またはアクティベーション時にインストールする必要がある NX-OS 機能 RPM が保持されます。リポジトリは、古いインストールからのすべての永続化された RPM を保持します。**localdblocaldb** に存在し、不要になった非アクティブな RPM はすべて削除して、新しい RPM 用のスペースを確保する必要があります。

YUM コマンドコマンドまたは **install** コマンドは、これらの RPM のインストールまたは削除に使用できます。

次のルールが、ブートまたはインストール時の機能 RPM のインストール手順に適用されます。

- NXOS リリースと同じリリース番号の RPM ファイルが、アクティブ化される互換性のあるファイルです。
- RPM が **groups-repo** リポジトリに存在する場合、ユーザーは RPM を **localdb** リポジトリに追加することはできません。 **groups-repo** リポジトリ以外の RPM バージョンを許可する必要があります。
- RPM が **groups-repo** リポジトリに存在し、同じバージョンの **localdb** リポジトリにも存在する場合、 **groups-repo** リポジトリからの RPM は、ブート時およびインストール時にインストールに考慮されます。（古いリリースでは、ベース RPM を追加すると **localdb** リポジトリへのインストールが許可されていたため、この手順が必要でしたこの手順は、下位互換性のために必要です。）
- 非基本機能の RPM が追加、アクティブ化、非アクティブ化、または削除されると、それぞれのエントリが **inactive\_feature\_rpms.inf** に存在する必要があります。
- ベース RPM が非アクティブ化またはアクティブ化されている場合、エントリはそれぞれ **/bootflash/.rpmstore/nxos\_preinstall\_rpms\_removed** および **/bootflash/.rpmstore/nxos\_rpms\_persisted** に存在する必要があります。
- ベース RPM エントリが **/bootflash/.rpmstore/nxos\_rpms\_persisted** ファイルに存在しない場合、 **groups-repo** または **localdb** リポジトリに存在する RPM バージョンは、ブート時にインストールの対象として考慮されません。この場合、削除されたエントリは **/bootflash/.rpmstore/nxos\_preinstall\_rpms\_removed** の一部である必要があります。



- 
- (注) RPM をシステムのリポジトリに手動でコピーをすることは避けてください。代わりに、**install** または YUM コマンドを使用してください。
- 

## YUM コマンドに関する情報

YUM コマンドの詳細については、以降のセクションを参照してください。



- 
- (注) YUM コマンドは **ctrl+c** をサポートしていません。インストール コマンドは **ctrl+c** をサポートしています。YUM コマンドが **ctrl+c** を使用して中断された場合は、「**/isan/bin/patching\_utils.py --unlock**」を使用して手動でクリーンアップする必要があります。
-

## YUM コマンドを使用したパッケージの操作

YUM コマンドを使用してパッケージを操作するには、以下のセクションを参照してください。



(注) YUM コマンドは、ボックスの BASH シェルからのみアクセスできます。NXOS VSH ターミナルからはアクセスできません。



(注) sudo ユーザとして、スーパー ユーザ権限にアクセスできることを確認してください。

### イメージのベースバージョン RPM を特定する

**ls /rpms** コマンドを使用して、イメージのベースバージョン RPM を特定します。ベース RPM バージョンは、システム イメージにアーカイブされた、事前インストール済みの RPM です。

```
#ls /rpms
```

```
bfd-2.0.0.0-9.2.1.lib32_n9000.rpm
ins_tor_sdk_t2-1.0.0.0-9.2.0.77.lib32_n9000.rpm
mtx-netconf-agent-2.0.0.0-9.2.1.lib32_n9000.rpm  snmp-2.0.0.0-9.2.1.lib32_n9000.rpm
bgp-2.0.0.0-9.2.1.lib32_n9000.rpm
ins_tor_sdk_t3-1.0.0.0-9.2.0.77.lib32_n9000.rpm
mtx-restconf-agent-2.0.0.0-9.2.1.lib32_n9000.rpm  sr-2.0.0.0-9.2.1.lib32_n9000.rpm
container-tracker-2.0.0.0-9.2.1.lib32_n9000.rpm  isis-2.0.0.0-9.2.1.lib32_n9000.rpm
mtx-telemetry-2.0.0.0-9.2.1.lib32_n9000.rpm
svi-2.0.0.0-9.2.1.lib32_n9000.rpm
eigrp-2.0.0.0-9.2.1.lib32_n9000.rpm  lACP-2.0.0.0-9.2.1.lib32_n9000.rpm
nbproxy-2.0.0.0-9.2.1.lib32_n9000.rpm
tacacs-2.0.0.0-9.2.1.lib32_n9000.rpm
ext-eth-2.0.0.0-9.2.1.lib32_n9000.rpm  lldp-2.0.0.0-9.2.1.lib32_n9000.rpm
ntp-2.0.0.0-9.2.1.lib32_n9000.rpm
telemetry-2.3.4.0-9.2.1.lib32_n9000.rpm
fcoe-2.0.0.0-9.2.1.lib32_n9000.rpm  mcast-2.0.0.0-9.2.1.lib32_n9000.rpm
nxos-ssh-2.0.0.0-9.2.1.lib32_n9000.rpm
virtualization-2.0.0.0-9.2.1.lib32_n9000.rpm
fex-2.0.0.0-9.2.1.lib32_n9000.rpm  mpls-2.0.0.0-9.2.1.lib32_n9000.rpm
ospf-2.0.0.0-9.2.1.lib32_n9000.rpm
vtp-2.0.0.0-9.2.1.lib32_n9000.rpm
fhrp-2.0.0.0-9.2.1.lib32_n9000.rpm  mtX-device-2.0.0.0-9.2.1.lib32_n9000.rpm
repdata
vxlan-2.0.0.0-9.2.1.lib32_n9000.rpm
guestshell-2.0.0.0-9.2.1.lib32_n9000.rpm
mtx-grpc-agent-2.0.0.0-9.2.1.lib32_n9000.rpm  rip-2.0.0.0-9.2.1.lib32_n9000.rpm
icam-2.0.0.0-9.2.1.lib32_n9000.rpm  mtX-infra-2.0.0.0-9.2.1.lib32_n9000.rpm
services-2.0.0.0-9.2.1.lib32_n9000.rpm
```

### インストール済み RPM のリストをチェックする

**yum list installed** コマンドを使用して機能 RPM とサードパーティ RPM をクエリして、特定の RPM を **grep** 検索します。機能 RPM については、次の例を参照してください。

```

bash-4.2# yum list installed | grep lib32_n9000

bfd.lib32_n9000                2.0.0.0-9.2.1      @groups-repo
core.lib32_n9000              2.0.0.0-9.2.1      installed
eth.lib32_n9000              2.0.0.0-9.2.1      installed
guestshell.lib32_n9000       2.0.0.0-9.2.1      @groups-repo
lACP.lib32_n9000             2.0.0.0-9.2.1      installed
linecard2.lib32_n9000        2.0.0.0-9.2.1      installed
lldp.lib32_n9000             2.0.0.0-9.2.1      installed
mcast.lib32_n9000            2.0.0.0-9.2.1      @groups-repo
mtx-device.lib32_n9000        2.0.0.0-9.2.1      installed
mtx-grpc-agent.lib32_n9000    2.0.0.0-9.2.1      installed
mtx-infra.lib32_n9000         2.0.0.0-9.2.1      installed
mtx-netconf-agent.lib32_n9000 2.0.0.0-9.2.1      installed
mtx-restconf-agent.lib32_n9000 2.0.0.0-9.2.1      installed
mtx-telemetry.lib32_n9000    2.0.0.0-9.2.1      installed
nbproxy.lib32_n9000          2.0.0.0-9.2.1      installed
ntp.lib32_n9000              2.0.0.0-9.2.1      installed
nxos-ssh.lib32_n9000         2.0.0.0-9.2.1      installed
ospf.lib32_n9000             2.0.0.0-9.2.1      @groups-repo
platform.lib32_n9000         2.0.0.0-9.2.1      installed
snmp.lib32_n9000             2.0.0.0-9.2.1      installed
svi.lib32_n9000              2.0.0.0-9.2.1      installed
tacacs.lib32_n9000           2.0.0.0-9.2.1      installed
tor.lib32_n9000              2.0.0.0-9.2.0.77   installed
virtualization.lib32_n9000    2.0.1.0-9.2.1      @localdb
vtp.lib32_n9000              2.0.0.0-9.2.1      installed
vxlan.lib32_n9000            2.0.0.0-9.2.1      @groups-repo
...

```

## インストール済み RPM の詳細を取得する

**yum info <rpmname>** コマンドは、インストール済み RPM の詳細情報リストを出力します。

### yum info vxlan

```

Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
groups-repo

```

```

localdb | 1.1 kB 00:00 ...

patching | 951 B 00:00 ...

thirdparty | 951 B 00:00 ...

Installed Packages | 951 B 00:00 ...

```

```

Installed Packages
Name      : vxlan
Arch      : lib32_n9000
Version   : 2.0.0.0
Release  : 9.2.1
Size      : 6.4 M
Repo      : installed
From repo : groups-repo
Summary   : Cisco NXOS VxLAN
URL       : http://cisco.com/

```

```
License      : Proprietary
Description  : Provides VxLAN support
```

## RPM のインストール

RPM をインストールすると、RPM がダウンロードされ、それぞれのプログラムがスイッチにコピーされます。次の例は、RPM をリモートサーバ（ネットワークで到達可能）からインストールする場合を示しています。

```
bash-4.3# yum install
http://10.0.0.2/modularity/rpms/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm
```

```
Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
groups-repo
```

```
localdb | 1.1 kB 00:00 ...
localdb | 951 B 00:00 ...
localdb/primary | 886 B 00:00 ...
localdb | 1/1
patching | 951 B 00:00 ...
thirdparty | 951 B 00:00 ...
```

```
Setting up Install Process
```

```
vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm
```

```
| 1.6 MB 00:00
```

```
Examining /var/tmp/yum-root-RaANgb/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm:
```

```
vxlan-2.0.1.0-9.2.1.lib32_n9000
```

```
Marking /var/tmp/yum-root-RaANgb/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm to be installed
```

```
Resolving Dependencies
```

```
--> Running transaction check
```

```
---> Package vxlan.lib32_n9000 0:2.0.1.0-9.2.1 will be installed
```

```
--> Finished Dependency Resolution
```

```
Dependencies Resolved
```

| Package     | Repository                       | Arch        | Version       | Size  |
|-------------|----------------------------------|-------------|---------------|-------|
| Installing: |                                  |             |               |       |
| vxlan       | /vxlan-2.0.1.0-9.2.1.lib32_n9000 | lib32_n9000 | 2.0.1.0-9.2.1 | 6.4 M |

```
Transaction Summary
```

```
Install 1 Package
```

```
Total size: 6.4 M
```

```
Installed size: 6.4 M
```

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

```
Downloading Packages:
```

```
Running Transaction Check
```

```
Running Transaction Test
```

```
Transaction Test Succeeded
```

```
Running Transaction
```

```
Installing : vxlan-2.0.1.0-9.2.1.lib32_n9000
```

```
1/1
```

```

starting pre-install package version mgmt for vxlan
pre-install for vxlan complete
starting post-install package version mgmt for vxlan
post-install for vxlan complete

```

```

Installed:
  vxlan.lib32_n9000 0:2.0.1.0-9.2.1

```

```
Complete!
```

次の例は、RPM をローカル ブートフラッシュからインストールする場合を示しています。

```
sudo yum install /bootflash/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm
```

```

Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
groups-repo

```

```

localdb                | 1.1 kB    00:00 ...
patching                | 951 B     00:00 ...
thirdparty              | 951 B     00:00 ...

```

```
Setting up Install Process
```

```

Examining /bootflash/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm: vxlan-2.0.1.0-9.2.1.lib32_n9000
Marking /bootflash/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm as an update to
vxlan-2.0.0.0-9.2.1.lib32_n9000

```

```
Resolving Dependencies
```

```
--> Running transaction check
```

```
---> Package vxlan.lib32_n9000 0:2.0.0.0-9.2.1 will be updated
```

```
---> Package vxlan.lib32_n9000 0:2.0.1.0-9.2.1 will be an update
```

```
--> Finished Dependency Resolution
```

```
Dependencies Resolved
```

| Package<br>Version     | Arch<br>Size         | Repository                       |
|------------------------|----------------------|----------------------------------|
| vxlan<br>2.0.1.0-9.2.1 | lib32_n9000<br>6.4 M | /vxlan-2.0.1.0-9.2.1.lib32_n9000 |

```
Transaction Summary
```

```
Upgrade      1 Package
```

```
Total size: 6.4 M
```

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

```
Downloading Packages:
```

```
Running Transaction Check
```

```
Running Transaction Test
```

```
Transaction Test Succeeded
```

```
Running Transaction
```

```

Updating      : vxlan-2.0.1.0-9.2.1.lib32_n9000
                1/2
starting pre-install package version mgmt for vxlan
pre-install for vxlan complete
starting post-install package version mgmt for vxlan
post-install for vxlan complete
Cleanup      : vxlan-2.0.0.0-9.2.1.lib32_n9000
                2/2

Updated:
  vxlan.lib32_n9000 0:2.0.1.0-9.2.1

```

Complete!

次の例は、RPM がリポジトリ内で利用可能な場合の RPM のインストールを示しています。

```
yum install eigrp
```

## RPM のアップグレード

次の例は、RPM をリモート サーバ（ネットワークで到達可能）からアップグレードする場合を示しています。

```

bash-4.3# yum upgrade
http://10.0.0.2/modularity/rpms/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm

Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
groups-repo
                | 1.1 kB    00:00 ...
localdb
                | 951 B    00:00 ...
patching
                | 951 B    00:00 ...
thirdparty
                | 951 B    00:00 ...

Setting up Upgrade Process
vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm
                | 1.6 MB    00:00
Examining /var/tmp/yum-root-RaANgb/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm:
vxlan-2.0.1.0-9.2.1.lib32_n9000
Marking /var/tmp/yum-root-RaANgb/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm as an update to
vxlan-2.0.0.0-9.2.1.lib32_n9000
Resolving Dependencies
--> Running transaction check
---> Package vxlan.lib32_n9000 0:2.0.0.0-9.2.1 will be updated
---> Package vxlan.lib32_n9000 0:2.0.1.0-9.2.1 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

```

| Package | Repository | Arch | Version | Size |
|---------|------------|------|---------|------|
|---------|------------|------|---------|------|

```

Updating:
vxlan                lib32_n9000                2.0.1.0-9.2.1
                   /vxlan-2.0.1.0-9.2.1.lib32_n9000 6.4 M
Transaction Summary
-----
Upgrade            1 Package

Total size: 6.4 M
Is this ok [y/N]: y
Downloading Packages:
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
** Found 1 pre-existing rpmdb problem(s), 'yum check' output follows:
busybox-1.23.2-r0.0.x86_64 has missing requires of busybox-syslog
  Updating      : vxlan-2.0.1.0-9.2.1.lib32_n9000                1/2

starting pre-install package version mgmt for vxlan
pre-install for vxlan complete
starting post-install package version mgmt for vxlan
post-install for vxlan complete
  Cleanup      : vxlan-2.0.0.0-9.2.1.lib32_n9000                2/2

Updated:
  vxlan.lib32_n9000 0:2.0.1.0-9.2.1

```

Complete!

次の例は、RPM をローカルブートフラッシュからアップグレードする場合を示しています。

```
sudo yum upgrade /bootflash/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm
```

```

Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
groups-repo

localdb                | 1.1 kB    00:00 ...
patching               | 951 B     00:00 ...
thirdparty             | 951 B     00:00 ...

Setting up Upgrade Process
Examining /bootflash/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm: vxlan-2.0.1.0-9.2.1.lib32_n9000
Marking /bootflash/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm as an update to
vxlan-2.0.0.0-9.2.1.lib32_n9000
Resolving Dependencies
--> Running transaction check
---> Package vxlan.lib32_n9000 0:2.0.0.0-9.2.1 will be updated
---> Package vxlan.lib32_n9000 0:2.0.1.0-9.2.1 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

```

```

Package                               Arch                               Repository
Version                               Size
-----
Updating:
vxlan                                  lib32_n9000                       /vxlan-2.0.1.0-9.2.1.lib32_n9000
  2.0.1.0-9.2.1                        6.4 M

Transaction Summary
-----
Upgrade      1 Package

Total size: 6.4 M
Is this ok [y/N]: y
Downloading Packages:
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Updating   : vxlan-2.0.1.0-9.2.1.lib32_n9000

                               1/2
starting pre-install package version mgmt for vxlan
pre-install for vxlan complete
starting post-install package version mgmt for vxlan
post-install for vxlan complete
  Cleanup   : vxlan-2.0.0.0-9.2.1.lib32_n9000

                               2/2

Updated:
  vxlan.lib32_n9000 0:2.0.1.0-9.2.1

Complete!

```

次の例は、RPM がリポジトリ内で利用可能な場合の RPM のアップグレードを示しています。

```
yum upgrade eigrp
```

## RPM のダウングレード

次の例は、RPM をリモート サーバ（ネットワークで到達可能）からダウングレードする場合を示しています。

```
sudo yum downgrade vxlan-2.0.0.0-9.2.1.lib32_n9000
```

```

Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
Setting up Downgrade Process
groups-repo

localdb                               | 1.1 kB    00:00 ...

```

```

localdb/primary          | 951 B    00:00 ...
localdb                  | 1.3 kB   00:00 ...
                          2/2
patching
thirdparty              | 951 B    00:00 ...
                          | 951 B    00:00 ...
Resolving Dependencies
--> Running transaction check
---> Package vxlan.lib32_n9000 0:2.0.0.0-9.2.1 will be a downgrade
---> Package vxlan.lib32_n9000 0:2.0.1.0-9.2.1 will be erased
--> Finished Dependency Resolution

```

Dependencies Resolved

| Package      | Version       | Size  | Arch        | Repository  |
|--------------|---------------|-------|-------------|-------------|
| Downgrading: |               |       |             |             |
| vxlan        | 2.0.0.0-9.2.1 | 1.6 M | lib32_n9000 | groups-repo |

Transaction Summary

Downgrade 1 Package

Total download size: 1.6 M

Is this ok [y/N]: y

Downloading Packages:

Running Transaction Check

Running Transaction Test

Transaction Test Succeeded

Running Transaction

Installing : vxlan-2.0.0.0-9.2.1.lib32\_n9000

1/2

starting pre-install package version mgmt for vxlan

pre-install for vxlan complete

starting post-install package version mgmt for vxlan

post-install for vxlan complete

Cleanup : vxlan-2.0.1.0-9.2.1.lib32\_n9000

2/2

Removed:

vxlan.lib32\_n9000 0:2.0.1.0-9.2.1

Installed:

vxlan.lib32\_n9000 0:2.0.0.0-9.2.1

```
Complete!
```

次の例は、RPM をローカルブートフラッシュからダウングレードする場合を示しています。

```
yum downgrade /bootflash/eigrp-2.0.0-9.2.1.lib32_n9000.rpm
```

次の例は、RPM がいずれかのリポジトリ内で利用可能な場合のRPMのダウングレードを示しています。

```
yum downgrade eigrp
```

## RPM の削除

RPM を削除すると、RPM がアンインストールされ、機能のコンフィグレーション CLI がすべて削除されます。RPM を削除するには、**yum erase <rpm>** コマンドを使用します。

```
bash-4.2# sudo yum erase vxlan
```

```
Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
Setting up Remove Process
Resolving Dependencies
--> Running transaction check
---> Package vxlan.lib32_n9000 0:2.0.1.0-9.2.1 will be erased
--> Finished Dependency Resolution
```

```
Dependencies Resolved
```

| Package             | Arch        | Repository                        | Version       |
|---------------------|-------------|-----------------------------------|---------------|
| Size                |             |                                   |               |
| Removing:           |             |                                   |               |
| vxlan               | lib32_n9000 | @/vxlan-2.0.1.0-9.2.1.lib32_n9000 | 2.0.1.0-9.2.1 |
| 6.4 M               |             |                                   |               |
| Transaction Summary |             |                                   |               |

```
Remove      1 Package
```

```
Installed size: 6.4 M
Is this ok [y/N]: y
Downloading Packages:
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Erasing      : vxlan-2.0.1.0-9.2.1.lib32_n9000

                               1/1
starting pre-remove package version mgmt for vxlan
pre-remove for vxlan complete
```

```
Removed:
```

```
vxlan.lib32_n9000 0:2.0.1.0-9.2.1
```

```
Complete!
```

## YUM グループのサポート

YUM グループのサポートは、パッケージ管理の一部です。これにより、管理者によるパッケージの管理が簡素化され、高度な柔軟性が提供されます。

管理者は、パッケージ (RPM) のリストを論理グループにグループ化して、さまざまな操作を実行することができます。YUM では、次の `group` コマンドがサポートされています。

- `grouplist`
- `groupinfo`
- `groupinstall`
- `groupremove`
- `groupupdate`

YUM グループは、L2、L3、ルーティング、および管理として幅広く分類が可能です。

### `grouplist` コマンドを使用する

Linux では、複数のパッケージを特定のグループにまとめることができます。yum でパッケージを個別にインストールするのではなく、特定のグループをインストールして、そのグループに属するすべての関連パッケージをインストールできます。たとえば、使用可能なグループをすべてリストするには `yum grouplist` コマンドを使用します。

```
bash-4.2# sudo yum grouplist
```

```
Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
Setting up Group Process
groups-repo
```

```
localdb | 1.1 kB 00:00 ...
patching | 951 B 00:00 ...
thirdparty | 951 B 00:00 ...
groups-repo/group | 951 B 00:00 ...
Installed Groups: | 1.6 kB 00:00 ...
  L2
  L3
  management
```

**groupmembers** コマンドを使用する

```
Available Groups:
  routing
Done
bash-4.3$
```

**groupmembers** コマンドを使用する

あるパッケージグループの説明と内容を表示するには、**yum groupinfo** コマンドを使用します。このコマンドは、グループの機能メンバのリストを出力します。

```
bash-4.2# sudo yum groupinfo 12

Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
Setting up Group Process
groups-repo

localdb                | 1.1 kB    00:00 ...
patching               | 951 B     00:00 ...
thirdparty             | 951 B     00:00 ...
                       | 951 B     00:00 ...

Group: L2
Mandatory Packages:
  lacp
  lldp
  svi
  vtp
```

**groupinstall** コマンドを使用する

このコマンドは、メンバ RPM のインストールとアップグレードの両方に使用します。メンバがインストールされていない場合は、使用可能な最も高いバージョンがインストールされます。メンバがすでにインストールされていてより高いバージョンの RPM が使用可能である場合、このコマンドでそのメンバがアップグレードされます。

```
bash-4.2# sudo yum groupinstall routing

Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
groups-repo

localdb                | 1.1 kB    00:00 ...
patching               | 951 B     00:00 ...
```

```

thirdparty | 951 B 00:00 ...

Setting up Group Process
Package ospf-2.0.0.0-9.2.1.lib32_n9000 already installed and latest version
Resolving Dependencies
--> Running transaction check
---> Package bgp.lib32_n9000 0:2.0.0.0-9.2.1 will be installed
---> Package eigrp.lib32_n9000 0:2.0.0.0-9.2.1 will be installed
---> Package isis.lib32_n9000 0:2.0.0.0-9.2.1 will be installed
---> Package rip.lib32_n9000 0:2.0.0.0-9.2.1 will be installed
--> Finished Dependency Resolution

```

Dependencies Resolved

| Package     | Arch        | Repository  | Version<br>Size        |
|-------------|-------------|-------------|------------------------|
| Installing: |             |             |                        |
| bgp         | lib32_n9000 | groups-repo | 2.0.0.0-9.2.1<br>2.4 M |
| eigrp       | lib32_n9000 | groups-repo | 2.0.0.0-9.2.1<br>428 k |
| isis        | lib32_n9000 | groups-repo | 2.0.0.0-9.2.1<br>1.2 M |
| rip         | lib32_n9000 | groups-repo | 2.0.0.0-9.2.1<br>214 k |

Transaction Summary

Install 4 Packages

Total download size: 4.2 M

Installed size: 19 M

Is this ok [y/N]: y

Downloading Packages:

Total

132 MB/s | 4.2 MB 00:00

Running Transaction Check

Running Transaction Test

Transaction Test Succeeded

Running Transaction

Installing : rip-2.0.0.0-9.2.1.lib32\_n9000

1/4

starting pre-install package version mgmt for rip

pre-install for rip complete

starting post-install package version mgmt for rip

post-install for rip complete

Installing : isis-2.0.0.0-9.2.1.lib32\_n9000

2/4

starting pre-install package version mgmt for isis

pre-install for isis complete

starting post-install package version mgmt for isis

post-install for isis complete

Installing : eigrp-2.0.0.0-9.2.1.lib32\_n9000

3/4

starting pre-install package version mgmt for eigrp

## groupupdate コマンドを使用する

```

pre-install for eigrp complete
starting post-install package version mgmt for eigrp
post-install for eigrp complete
  Installing : bgp-2.0.0.0-9.2.1.lib32_n9000

                                         4/4
starting pre-install package version mgmt for bgp
pre-install for bgp complete
starting post-install package version mgmt for bgp
post-install for bgp complete

Installed:
  bgp.lib32_n9000 0:2.0.0.0-9.2.1          eigrp.lib32_n9000 0:2.0.0.0-9.2.1
             isis.lib32_n9000 0:2.0.0.0-9.2.1          rip.lib32_n9000
0:2.0.0.0-9.2.1

Complete!

```

## groupupdate コマンドを使用する

既存のインストール済みグループパッケージをすべて更新するには、**yum groupupdate** コマンドを使用します。

```

bash-4.3# yum groupupdate routing

Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
groups-repo

localdb                | 1.1 kB    00:00 ...
localdb/primary        | 951 B     00:00 ...
localdb                | 1.9 kB    00:00 ...
patching               6/6
thirdparty            | 951 B     00:00 ...
                       | 951 B     00:00 ...

Setting up Group Process
Resolving Dependencies
--> Running transaction check
---> Package bgp.lib32_n9000 0:2.0.0.0-9.2.1 will be updated
---> Package bgp.lib32_n9000 0:2.0.1.0-9.2.1 will be an update
---> Package eigrp.lib32_n9000 0:2.0.0.0-9.2.1 will be updated
---> Package eigrp.lib32_n9000 0:2.0.1.0-9.2.1 will be an update
---> Package isis.lib32_n9000 0:2.0.0.0-9.2.1 will be updated
---> Package isis.lib32_n9000 0:2.0.1.0-9.2.1 will be an update
---> Package ospf.lib32_n9000 0:2.0.0.0-9.2.1 will be updated
---> Package ospf.lib32_n9000 0:2.0.1.0-9.2.1 will be an update
---> Package rip.lib32_n9000 0:2.0.0.0-9.2.1 will be updated
---> Package rip.lib32_n9000 0:2.0.1.0-9.2.1 will be an update
--> Finished Dependency Resolution

```

Dependencies Resolved

| Package             | Arch        | Repository | Size | Version |
|---------------------|-------------|------------|------|---------|
| Updating:           |             |            |      |         |
| bgp                 | lib32_n9000 |            |      |         |
| 2.0.1.0-9.2.1       |             | localdb    |      | 2.4 M   |
| eigrp               | lib32_n9000 |            |      |         |
| 2.0.1.0-9.2.1       |             | locald     |      | 428 k   |
| isis                | lib32_n9000 |            |      |         |
| 2.0.1.0-9.2.1       |             | local      |      | 1.2 M   |
| ospf                | lib32_n9000 |            |      |         |
| 2.0.1.0-9.2.1       |             | localdb    |      | 2.8 M   |
| rip                 | lib32_n9000 |            |      |         |
| 2.0.1.0-9.2.1       |             | localdb    |      | 214 k   |
| Transaction Summary |             |            |      |         |

Upgrade 5 Packages

Total download size: 7.0 M

Is this ok [y/N]: y

Downloading Packages:

Total

269 MB/s | 7.0 MB 00:00

Running Transaction Check

Running Transaction Test

Transaction Test Succeeded

Running Transaction

Updating : eigrp-2.0.1.0-9.2.1.lib32\_n9000

1/10

starting pre-install package version mgmt for eigrp

pre-install for eigrp complete

starting post-install package version mgmt for eigrp

post-install for eigrp complete

Updating : ospf-2.0.1.0-9.2.1.lib32\_n9000

2/10

starting pre-install package version mgmt for ospf

pre-install for ospf complete

starting post-install package version mgmt for ospf

post-install for ospf complete

Updating : rip-2.0.1.0-9.2.1.lib32\_n9000

3/10

starting pre-install package version mgmt for rip

pre-install for rip complete

starting post-install package version mgmt for rip

post-install for rip complete

Updating : isis-2.0.1.0-9.2.1.lib32\_n9000

4/10

starting pre-install package version mgmt for isis

pre-install for isis complete

starting post-install package version mgmt for isis

post-install for isis complete

Updating : bgp-2.0.1.0-9.2.1.lib32\_n9000

## grouperase コマンドを使用する

```

5/10
starting pre-install package version mgmt for bgp
pre-install for bgp complete
starting post-install package version mgmt for bgp
post-install for bgp complete
Cleanup      : bgp-2.0.0.0-9.2.1.lib32_n9000

6/10
Cleanup      : isis-2.0.0.0-9.2.1.lib32_n9000

7/10
Cleanup      : rip-2.0.0.0-9.2.1.lib32_n9000

8/10
Cleanup      : ospf-2.0.0.0-9.2.1.lib32_n9000

9/10
Cleanup      : eigrp-2.0.0.0-9.2.1.lib32_n9000

10/10

Updated:
  bgp.lib32_n9000 0:2.0.1.0-9.2.1      eigrp.lib32_n9000 0:2.0.1.0-9.2.1
  isis.lib32_n9000 0:2.0.1.0-9.2.1    ospf.lib32_n9000 0:2.0.1.0-9.2.1
  rip.lib32_n9000 0:2.0.1.0-9.2.1

Complete!

```

## grouperase コマンドを使用する

グループ、またはグループのすべての RPM メンバを削除するには、**yum grouperase** コマンドを使用します。

```

bash-4.3$ sudo yum grouperase routing

Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
Setting up Group Process
groups-repo

localdb          | 1.1 kB    00:00 ...
patching        | 951 B     00:00 ...
thirdparty      | 951 B     00:00 ...

Resolving Dependencies
--> Running transaction check
---> Package bgp.lib32_n9000 0:2.0.0.0-9.2.1 will be erased
---> Package eigrp.lib32_n9000 0:2.0.0.0-9.2.1 will be erased
---> Package isis.lib32_n9000 0:2.0.0.0-9.2.1 will be erased
---> Package ospf.lib32_n9000 0:2.0.0.0-9.2.1 will be erased
---> Package rip.lib32_n9000 0:2.0.0.0-9.2.1 will be erased
--> Finished Dependency Resolution

Dependencies Resolved

```

| Package   | Arch        | Repository   | Version<br>Size        |
|-----------|-------------|--------------|------------------------|
| Removing: |             |              |                        |
| bgp       | lib32_n9000 | @groups-repo | 2.0.0.0-9.2.1<br>11 M  |
| eigrp     | lib32_n9000 | @groups-repo | 2.0.0.0-9.2.1<br>2.0 M |
| isis      | lib32_n9000 | @groups-repo | 2.0.0.0-9.2.1<br>5.7 M |
| ospf      | lib32_n9000 | @groups-repo | 2.0.0.0-9.2.1<br>15 M  |
| rip       | lib32_n9000 | @groups-repo | 2.0.0.0-9.2.1<br>1.0 M |

## Transaction Summary

Remove 5 Packages

Installed size: 34 M

Is this ok [y/N]: y

Downloading Packages:

Running Transaction Check

Running Transaction Test

Transaction Test Succeeded

Running Transaction

Erasing : isis-2.0.0.0-9.2.1.lib32\_n9000

1/5

starting pre-remove package version mgmt for isis

pre-remove for isis complete

Erasing : ospf-2.0.0.0-9.2.1.lib32\_n9000

2/5

starting post-remove package version mgmt for isis

post-remove for isis complete

starting pre-remove package version mgmt for ospf

pre-remove for ospf complete

Erasing : eigrp-2.0.0.0-9.2.1.lib32\_n9000

3/5

starting post-remove package version mgmt for ospf

post-remove for ospf complete

starting pre-remove package version mgmt for eigrp

pre-remove for eigrp complete

Erasing : rip-2.0.0.0-9.2.1.lib32\_n9000

4/5

starting post-remove package version mgmt for eigrp

post-remove for eigrp complete

starting pre-remove package version mgmt for rip

pre-remove for rip complete

Erasing : bgp-2.0.0.0-9.2.1.lib32\_n9000

5/5

starting post-remove package version mgmt for rip

post-remove for rip complete

starting pre-remove package version mgmt for bgp

pre-remove for bgp complete

Removed:

bgp.lib32\_n9000 0:2.0.0.0-9.2.1 eigrp.lib32\_n9000 0:2.0.0.0-9.2.1  
isis.lib32\_n9000 0:2.0.0.0-9.2.1 ospf.lib32\_n9000 0:2.0.0.0-9.2.1

```
rip.lib32_n9000 0:2.0.0.0-9.2.1
Complete!
```

## リポジトリを特定する

このコマンドは、スイッチに存在するリポジトリに加え、これらのリポジトリに含まれる RPM の数をリストします。

```
bash-4.3# yum repolist all
```

```
Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
groups-repo
```

|              |                         |             |
|--------------|-------------------------|-------------|
| localdb      | 1.1 kB                  | 00:00 ...   |
| patching     | 951 B                   | 00:00 ...   |
| thirdparty   | 951 B                   | 00:00 ...   |
| repo id      | 951 B                   | 00:00 ...   |
|              | repo name               | status      |
| groups-repo  | Groups-RPM Database     | enabled: 37 |
| localdb      | Local RPM Database      | enabled: 6  |
| patching     | Patch-RPM Database      | enabled: 0  |
| thirdparty   | Thirdparty RPM Database | enabled: 0  |
| open-nxos    | open-nxos               | disabled    |
| repolist: 43 |                         |             |

## インストール済み YUM バージョンを特定する

次に、インストール済み YUM バージョンをリストする例を示します。

```
yum --version
```

```
3.4.3
Installed: rpm-5.4.14-r0.0.x86_64 at 2018-06-02 13:04
Built      : Wind River <info@windriver.com> at 2018-04-27 08:36
Committed: Wind River <info@windriver.com> at 2018-04-27

Installed: yum-3.4.3-r9.0.x86_64 at 2018-06-02 13:05
Built      : Wind River <info@windriver.com> at 2018-04-27 08:36
Committed: Wind River <info@windriver.com> at 2018-04-27
```

## NX-OS CLI と YUM コマンドの対応関係

次の表は、NX-OS CLI とそれに対応する YUM コマンドを示しています。

表 6: パッチ適用コマンドリファレンス

| NX-OS CLI コマンド                       | YUM コマンド  |
|--------------------------------------|---|
| <code>show install inactive</code>   | <code>yum list --patch-only available</code>  |
| <code>show install active</code>     | <code>yum list --patch-only installed</code>  |
| <code>show install committed</code>  | <code>yum list --patch-only committed</code>  |
| <code>show install packages</code>   | <code>yum list --patch-only</code>  |
| <code>show install pkg-info</code>   | <code>yum info --patch-only</code>  |
| <code>show install log</code>        | <p><code>yum history --show-patch-log</code></p> <p>ここで <code>log_cmd</code> は次のとおりです。</p> <ul style="list-style-type: none"> <li>• <code>opid</code> : ある操作 ID に固有なログ</li> <li>• <code>last</code> : 最新の操作のログを示します。</li> <li>• <code>reverse</code> : 逆の順序でログを表示します。</li> <li>• <code>detail</code> : 詳細ログを表示します。</li> <li>• <code>from</code> : 特定の操作 ID 以降のログを示します。</li> </ul> |
| <code>clear install log</code>       | <p><code>yum history --clear-patch-log=</code></p> <p>ここで <code>clear_log_cmd</code> は次のとおりです。</p> <ul style="list-style-type: none"> <li>• <code>all</code> : ログを完全にクリアします。</li> <li>• <code>:</code> : この操作 ID 以降のログをクリアします。</li> </ul>   |
| <code>install add</code>             | <code>yum install --add bootflash:/</code>  |
| <code>install remove</code>          | <code>yum install --remove</code>   |
| <code>install remove inactive</code> | <code>yum install --remove all</code>   |
| <code>install activate</code>        | <p><code>yum install --no-persist --nocommit</code></p> <p>(注) デフォルトでは、すべてのパッケージがアクティブ化され、コミットされます。</p>   |

| NX-OS CLI コマンド                  | YUM コマンド   |
|---------------------------------|--|
| <code>install deactivate</code> | <code>yum erase --nocommit</code><br>(注) デフォルトでは、すべてのパッケージが非アクティブ化され、コミットされます。 |
| <code>install commit</code>     | <code>yum install --commit</code>  |
| <code>Install commit</code>     | <code>yum install --commit all</code>  |

## インストール操作ユーザ ロールの作成

`install` コマンドは、`admin` ロールのユーザのみが使用できます。`install` コマンドは、RBAC によりユーザが利用できるようになります。該当する RBAC 設定ガイドラインを参照してください。

## Compacting Cisco NX-OS ソフトウェア イメージの圧縮

Cisco NX-OS ソフトウェア画像の圧縮は、コピー要求を完了する前に画像ファイルのサイズを縮小します。送信元として SCP、HTTP、または HTTPS を使用し、宛先としてブートフラッシュまたは USB を使用します。次の例では SCP およびブートフラッシュを使用します。

```
switch# copy scp://user@scpserver.cisco.com/download/nxos.9.3.5.bin
bootflash:nxos.9.3.5.bin compact vrf management use-kstack
```

```
user1@10.65.42.196's password:
nxos.9.3.5.bin 100% 1887MB 6.6MB/s 04:47
Copy complete, now saving to disk (please wait)...
Copy complete.
```

ファイルをスーパーバイザ モジュールにコピーする前に、`compact` キーワードは NX-OS 画像を圧縮します。



(注) ソフトウェア画像の圧縮は、SCP、HTTP、または HTTPS でのみサポートされています。その他のプロトコルで圧縮しようとする場合、システムは次のエラーを返します。

```
Compact option is allowed only with source as scp/http/https and destination
as bootflash or usb
```



(注) 圧縮された画像は、LXC 起動モードではサポートされません。

## 翻訳について

このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。