



Cisco Nexus 3000 シリーズ NX-OS ソフトウェア アップグレード/ダウングレードガイド、リリース 9.3(x)

初版：2019年7月20日

最終更新：2021年2月22日

シスコシステムズ合同会社

〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー

<http://www.cisco.com/jp>

お問い合わせ先：シスコ コンタクトセンター

0120-092-255（フリーコール、携帯・PHS含む）

電話受付時間：平日 10:00～12:00、13:00～17:00

<http://www.cisco.com/jp/go/contactcenter/>

【注意】 シスコ製品をご使用になる前に、安全上の注意（ www.cisco.com/jp/go/safety_warning/ ）をご確認ください。本書は、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。また、契約等の記述については、弊社販売パートナー、または、弊社担当者にご確認ください。

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on standards documentation, or language that is used by a referenced third-party product.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/c/en/us/about/legal/trademarks.html>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2019–2021 Cisco Systems, Inc. All rights reserved.



目次

はじめに vii

対象読者 viii

表記法 ix

Cisco Nexus 9000 シリーズ スイッチの関連資料 xi

マニュアルに関するフィードバック xii

通信、サービス、およびその他の情報 xiii

第 1 章

新機能と変更情報 1

新機能と変更情報 1

第 2 章

Cisco Nexus 3000 シリーズ NX-OS ソフトウェアのアップグレードまたはダウングレード 3

ソフトウェア イメージの概要 4

サポート対象ハードウェア 4

ISSU について 5

ISSU の前提条件 5

ISSU の注意事項と制限事項 5

Cisco Nexus 3000、3100、および 3500 のコンパクトイメージ 7

シスコのソフトウェア ダウンロード Web サイトのコンパクト NX-OS ソフトウェア イメージ 7

異なるスイッチモデルで実行するコンパクトイメージ	8
NX-OS コンパクトイメージの手順	9
ブートフラッシュ/USBでのNX-OS コンパクトイメージの手順	9
SCPによるNX-OS コンパクトイメージの手順	9
NX-OS コンパクトイメージの手順シーケンス	9
NX-OS コンパクトイメージプラットフォームグループ	10
スイッチをUSBから起動する	11
BIOS およびパワー シーケンサ イメージのアップグレード	11
BIOS のアップグレード	11
各 Cisco Nexus 3000 シリーズ プラットフォームの BIOS バージョン	11
高速リロードシナリオでのアップグレードの注意事項	12
非高速リロードシナリオでのアップグレードのガイドライン	13
アップグレードの前提条件	13
アップグレード後の管理サービス	14
レイヤ2プロトコルの影響	14
スイッチのイーサネット インターフェイス	15
インストール前のチェック	15
高速リブートに関する情報	17
高速再起動のタイミング要件	17
高速リブートの 注意事項	17
Fast-Reload コマンドの使用	20
PSS/バイナリ構成での高速リロード	20
ASCII 構成での高速リロード	21
高速リロードのアップグレード	21
高速リブートによる BGP グレースフル リスタートを有効にする	22
高速リブートを使用したアップグレードとダウングレード	22
高速リブートを使用したアップグレード	22
高速リブートを使用したダウングレード	23
install all コマンドの使用	23
Install All Non-Interruptive コマンドの使用	24
アップグレード手順	24

インストールの概要	25
外部フラッシュ メモリ デバイスからの実行構成のコピー	28
外部フラッシュ メモリ デバイスからのスタートアップ構成のコピー	29
vPC 以外のトポロジでのアップグレードプロセス	30
vPC のアップグレードプロセス	30
プライマリ スイッチでの vPC トポロジのアップグレードプロセス	30
セカンダリ スイッチでの vPC トポロジのアップグレードプロセス	31
Nexus 9000 -RシリーズスイッチのvPCアップグレードおよびダウングレード手順	32
アップグレード ステータスのモニタリング	36
NX-OS アップグレード履歴	36
以前のリリースへのダウングレード	37
上位リリースから下位リリースへのダウングレード	38
インストールのトラブルシューティング	38

第 3 章

Cisco NX-OS ソフトウェアのオプション	39
Cisco NX-OS ソフトウェアのオプション	39
モジュラ パッケージの使用	41
Cisco NX-OS ソフトウェア パッケージのリスト	42
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	44
ISSU のサポート	44
RPM に関する情報	44
オプション RPM とその関連機能	44
NX-OS 機能 RPM インストールに関するガイドライン	46
パッチ適用可能な NX-OS 必須 RPM のリスト	48
機能 RPM の操作に install CLI を使用する	48
デジタル署名のサポートにインストール CLI を使用する	50
インストールされているすべての RPM のクエリ	51
インストール済み 機能 NX-OS RPM のみのクエリ	53
インストール済み サードパーティ RPM のみのクエリ	54
1 ステップ手順による RPM のインストール	55
2 ステップ手順による RPM のインストール	56

1 ステップの RPM のアップグレード	57
RPM のダウングレード	58
RPM の削除	59
RPM の形式	59
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	60
YUM コマンドに関する情報	61
YUM コマンドを使用したパッケージの操作	62
イメージのベース バージョン RPM を特定する	62
インストール済み RPM のリストをチェックする	62
インストール済み RPM の詳細を取得する	63
RPM のインストール	64
RPM のアップグレード	66
RPM のダウングレード	68
RPM の削除	70
YUM グループのサポート	71
リポジトリを特定する	78
インストール済み YUM バージョンを特定する	78
NX-OS CLI と YUM コマンドの対応関係	79
インストール操作ユーザ ロールの作成	80
Compacting Cisco NX-OS ソフトウェア イメージの圧縮	80

はじめに

この前書きは、次の項で構成されています。

対象読者

このマニュアルは、Cisco Nexus スイッチの設置、設定、および維持に携わるネットワーク管理者を対象としています。

表記法

コマンドの説明には、次のような表記法が使用されます。

表記法	説明
bold	太字の文字は、表示どおりにユーザが入力するコマンドおよびキーワードです。
<i>italic</i>	イタリック体の文字は、ユーザが値を指定する引数です。
[x]	省略可能な要素（キーワードまたは引数）は、角かっこで囲んで示しています。
[x y]	いずれか1つを選択できる省略可能なキーワードや引数は、角かっこで囲み、縦棒で区切って示しています。
{x y}	必ずいずれか1つを選択しなければならない必須キーワードや引数は、波かっこで囲み、縦棒で区切って示しています。
[x {y z}]	角かっこまたは波かっこが入れ子になっている箇所は、任意または必須の要素内の任意または必須の選択肢であることを表します。角かっこ内の波かっこと縦棒は、省略可能な要素内で選択すべき必須の要素を示しています。
variable	ユーザが値を入力する変数であることを表します。イタリック体を使用できない場合に使用されます。
string	引用符を付けない一組の文字。string の前後には引用符を使用しないでください。引用符を使用すると、その引用符も含めてstring と見なされます。

例では、次の表記法を使用しています。

表記法	説明
screen フォント	スイッチが表示する端末セッションおよび情報は、スクリーンフォントで示しています。
太字の screen フォント	ユーザが入力しなければならない情報は、太字の screen フォントで示しています。
イタリック体の screen フォント	ユーザが値を指定する引数は、イタリック体の screen フォントで示しています。
<>	パスワードのように出力されない文字は、山カッコ (<>) で囲んで示しています。
[]	システムプロンプトに対するデフォルトの応答は、角カッコで囲んで示しています。

表記法	説明
!, #	コードの先頭に感嘆符 (!) またはポンド記号 (#) がある場合には、コメント行であることを示します。

Cisco Nexus 9000 シリーズ スイッチの関連資料

Cisco Nexus 9000 シリーズ スイッチ全体のマニュアルセットは、次の URL にあります。

http://www.cisco.com/en/US/products/ps13386/tsd_products_support_series_home.html

マニュアルに関するフィードバック

このマニュアルに関する技術的なフィードバック、または誤りや記載もれなどお気づきの点がございましたら、HTML ドキュメント内のフィードバック フォームよりご連絡ください。ご協力をよろしくお願いいたします。

通信、サービス、およびその他の情報

- シスコからタイムリーな関連情報を受け取るには、[Cisco Profile Manager](#) でサインアップしてください。
- 重要な技術によりビジネスに必要な影響を与えるには、[Cisco Services](#) にアクセスしてください。
- サービス リクエストを送信するには、[Cisco Support](#) にアクセスしてください。
- 安全で検証済みのエンタープライズクラスのアプリケーション、製品、ソリューション、およびサービスを探して参照するには、[Cisco Marketplace](#) にアクセスしてください。
- 一般的なネットワーキング、トレーニング、認定関連の出版物を入手するには、[Cisco Press](#) にアクセスしてください。
- 特定の製品または製品ファミリの保証情報を探すには、[Cisco Warranty Finder](#) にアクセスしてください。

Cisco バグ検索ツール

[Cisco Bug Search Tool](#) (BST) は、シスコ製品とソフトウェアの障害と脆弱性の包括的なリストを管理する Cisco バグ追跡システムへのゲートウェイとして機能する、Web ベースのツールです。BST は、製品とソフトウェアに関する詳細な障害情報を提供します。



第 1 章

新機能と変更情報

この章では、『Cisco Nexus 3000 シリーズ NX-OS ソフトウェア アップグレードおよびダウングレードガイド、リリース 9.3(x)』に記載されている新しい機能と変更された機能に関するリリース固有の情報について説明します。

- [新機能と変更情報 \(1 ページ\)](#)

新機能と変更情報

表 1: NX-OS リリース 9.3(x) の新機能および変更された機能

特長	説明	変更が行われたリリース	参照先
Cisco NX-OS ソフトウェア イメージの圧縮	コピー操作中のソフトウェアイメージ圧縮のサポート。	9.3(5)	インストールの概要 (25 ページ) Cisco NX-OS ソフトウェアのオプション (39 ページ)
NX-OS アップグレード履歴	アップグレード全体でソフトウェアアップグレード履歴を維持するためのサポート。	9.3(5)	NX-OS アップグレード履歴 (36 ページ)
ISSU	Cisco Nexus 3016Q、3048TP、3064PQ、3064PQ-E、3064PQ-X、3064TQ、3132Q、3132Q-X、3172PQ、および 3172TQ プラットフォームスイッチのサポートが削除されました。	9.3(1)	ISSU の注意事項と制限事項 (5 ページ)

特長	説明	変更が行われたリリース	参照先
ソフトウェアアップグレード	Cisco NX-OS リリース 9.3(1) へのサポートされているアップグレードパスが追加されました。	9.3(1)	ISSU の注意事項と制限事項 (5 ページ)



第 2 章

Cisco Nexus 3000 シリーズ NX-OS ソフトウェアのアップグレードまたはダウングレード

このマニュアルでは、Cisco MDS 3000 シリーズ スイッチ用 Cisco NX-OS ソフトウェアのアップグレードまたはダウングレードの方法について説明します。

- [ソフトウェア イメージの概要 \(4 ページ\)](#)
- [サポート対象ハードウェア \(4 ページ\)](#)
- [ISSU について \(5 ページ\)](#)
- [Cisco Nexus 3000、3100、および 3500 のコンパクトイメージ \(7 ページ\)](#)
- [スイッチを USB から起動する \(11 ページ\)](#)
- [BIOS および パワー シーケンサ イメージのアップグレード \(11 ページ\)](#)
- [高速リロード シナリオでのアップグレードの注意事項 \(12 ページ\)](#)
- [非高速リロード シナリオでのアップグレードのガイドライン \(13 ページ\)](#)
- [アップグレードの前提条件 \(13 ページ\)](#)
- [インストール前のチェック \(15 ページ\)](#)
- [高速リブートに関する情報 \(17 ページ\)](#)
- [Fast-Reload コマンドの使用 \(20 ページ\)](#)
- [高速リブートを使用したアップグレードとダウングレード \(22 ページ\)](#)
- [アップグレード手順 \(24 ページ\)](#)
- [vPC のアップグレードプロセス \(30 ページ\)](#)
- [アップグレード ステータスのモニタリング \(36 ページ\)](#)
- [NX-OS アップグレード履歴 \(36 ページ\)](#)
- [以前のリリースへのダウングレード \(37 ページ\)](#)
- [インストールのトラブルシューティング \(38 ページ\)](#)

ソフトウェアイメージの概要

Cisco Nexus 3000 シリーズスイッチは、Cisco NX-OS ソフトウェアがスイッチに事前インストールされて出荷されます。既存のイメージをアップグレードまたはダウングレードする前に、このマニュアルの情報を最後までお読みになり、ソフトウェアのアップグレードに関する注意事項、前提条件、および手順について参照してください。Cisco Nexus 3000 シリーズスイッチの Cisco NX-OS ソフトウェアの更新情報については、『[Cisco Nexus 3000 Series Release Notes](#)』を参照してください。

Cisco NX-OS ソフトウェアは、1 つの Cisco NX-OS ソフトウェア イメージで構成されています。イメージのファイル名は、「nxos」から始まります。Cisco NX-OS オペレーティングシステムをロードするために必要なのは、このイメージだけです。このイメージは、すべての Cisco Nexus 3000 および 3100 シリーズスイッチで実行されます。**install all** コマンドの使用は、新しいリリースへのアップグレードでサポートされている唯一の方法です。



(注) Cisco Nexus 3000 シリーズスイッチをブートするための単一のイメージがあります。キックスタートイメージとシステムイメージは必要ありません。単一のイメージバイナリのため、次のコマンドが更新されます。

- `boot nxos single_image_binary`
- `install all nxos single_image_binary`

サポート対象ハードウェア

Cisco Nexus 3000 および 3100 シリーズスイッチは、Cisco NX-OS ソフトウェアが事前インストールされて出荷されます。Cisco NX-OS のアップグレードおよびダウングレードは、次の項に記載するハードウェアでサポートされています。

Cisco Nexus 3000 シリーズスイッチ

- Cisco Nexus 3016 スイッチ
- Cisco Nexus 3048 スイッチ
- Cisco Nexus 3064 スイッチ

Cisco Nexus 3100 シリーズスイッチ

- Cisco Nexus 3132Q および 3132Q-XL スイッチ
- Cisco Nexus 3172 スイッチ



- (注) N9K モードで動作する Cisco Nexus 3000 シリーズ スイッチのソフトウェア アップグレードおよびダウングレード情報については、『[Cisco Nexus 9000 Series NX-OS Software Upgrade and Downgrade Guide](#)』を参照してください。

ISSU について

in-service software upgrade (ISSU) は、スイッチがトラフィックの転送を続けながら、デバイスのソフトウェアをアップグレードすることができます。ISSU を使用すると、ソフトウェアのアップグレードによるダウンタイムを短縮するかゼロにすることができます。一部のスイッチでは、ISSU を実行できます。（サポートされているプラットフォームのリストについては、「ISSU の注意事項と制限事項」を参照してください）。

ISSU の前提条件

「ISSU の注意事項と制限事項」セクションの注意事項に従って、ISSU がスムーズに機能するようにします。

ISSU が進行している間は、ネットワークが安定しており変更が行われていないことを確認してください。

現在実行中のリリースとターゲットのリリースとの間の、機能の互換性も確認してください。

スパニングツリーで指定されたフォワーディングステートのインターフェイスがないことを確認してください。また、ブリッジアシュアランスがどのインターフェイスにも設定されていないことを確認してください。vPC ピアリンクは、これらの要件から除外されます。

現在の STP トポロジが ISSU の要件と整合性を維持しているかどうか確認します。 **show spanning-tree issu-impact** コマンドを使用すると、STP 設定と、STP に関する問題が発生する可能性があるかどうかが表示されます。

show lacp issu-impact コマンドを使用すると、いずれかのポートまたはピア スイッチがレート高速モードで設定されているかどうかが表示されます。システムが LACP 高速タイマーを構成している場合、ISSU プロセスは中断されます。

Cisco NX-OS リリース 9.3(1) に対して ISSU を実行する前に、Cisco Nexus 3132Q-XL、3172PQ-XL、および 3172TQ-XL プラットフォーム スイッチの BGP グレースフルリスタートタイマーを 180 秒に構成する必要があります。

ISSU の注意事項と制限事項

以下は、ISSU の重要な注意事項と制限事項のリストです。

- Cisco Nexus 3172PQ、3172TQ、3132Q、3132Q-X、3064、3064-X、3064-T、3048、3016 (4 GB の低メモリ プラットフォーム) での無停止の標準 ISSU は、Cisco Nexus 9.3(1) 以降のリリースではサポートされていません。

- Cisco Nexus 3132Q-XL、3172PQ-XL、および 3172TQ-XL スイッチは、Cisco NX-OS リリース 9.3(1) 以降のリリースへの ISSU をサポートします。詳細については、[ISSU サポートマトリックス](#)を参照してください。
- Cisco NX-OS リリース 7.0(3)I5(1) 以降、16 GB 以上のメモリを搭載した Cisco Nexus 3000 シリーズプラットフォームは拡張 ISSU をサポートします。アップグレードは中断を伴います。
- 以前のリリースから Cisco NX-OS リリース 9.3(1) へのフォールバック モードの拡張 ISSU はサポートされていません。Cisco NX-OS リリース 9.3(1) 以降、Cisco Nexus 3164Q、C31108PC-V、C31108TC-V、C3132Q-V は、フォールバック モードでのみ拡張 ISSU をサポートします。詳細については、「[ISSU サポートマトリックス](#)」を参照してください。
- Cisco Nexus 3264C-E スイッチは、Cisco NX-OS リリース 9.3(1) から将来の 9.3(x) リリースへの拡張 ISSU モード フォールバック機能をサポートしていません。
- ISSU を使用して 9.3(5) にアップグレードする以前のリリースバージョンでは、次のロギングレベルコマンドが設定されている場合、それらはアップグレードされたバージョンにないため、再設定する必要があります。
 - **logging level evmc value**
 - **logging level mvsh value**
 - **logging level fs-daemon value**
- ISSU を使用して Cisco NX-OS リリース 9.3(6) にアップグレードする以前のリリースバージョンでは、次のロギングレベルコマンドが設定されている場合、それらはアップグレードされたバージョンにないため、再設定する必要があります。
 - **logging level evmc value**
 - **logging level mvsh value**
- セグメントルーティングまたは VXLAN の ISSU を実行できます。
- ISSU を実行するために必要な最小空きブートフラッシュスペースは 550 MB です。
- CLI および SNMP の構成の変更要求は、ISSU 中に拒否されます。
- ISSU 中に STP トポロジの変更は想定されていません。
- ゲストシェルは、ISSU 実行中に無効化され、アップグレード後に再度アクティブ化されます。ISSU の間、ISSU の対象のノードがアクティブであると、すべてのファーストホップ冗長プロトコル (FHRP) は他のピアをアクティブにします。
- **boot poap enable** が構成されている場合、ISSU はブロックされます。
- Cisco NX-OS リリース 9.3(1)、9.3(2) または 9.3(3) から、Embedded Event Manager (EEM) 設定が実行設定に保存されている上位リリースにアップグレードすると、DME エラーが表示されることがあります。エラーは、**show consistency-checker dme running-config enhanced** コマンド、特にイベントマネージャ コマンドの出力にあります。このエラーが発生した

場合は、ISSU の完了後にすべての EEM アプレット構成を削除してから、EEM 構成を再適用します。

- モデル番号の末尾が **-XL** でない Cisco Nexus 3048、3064、3132、および 3172 スイッチは、ブートフラッシュ スペースが限られているため、コンパクトな NX-OS ソフトウェア イメージを実行する必要があります。このコンパクトイメージは、NX-OS コンパクトイメージの手順を使用して作成できます。または、コンパクト NX-OS ソフトウェア イメージを [シスコのソフトウェア ダウンロード Web サイト](#) から直接ダウンロードすることもできます。この要件は、Nexus 3000 または 3100 シリーズ スイッチの他のモデルには適用されません。
 - シスコの [ソフトウェアダウンロード Web サイト](#) で公開されているコンパクト NX-OS ソフトウェア イメージの MD5/SHA512 チェックサムは、NX-OS コンパクトイメージの手順で作成されたコンパクト イメージの MD5/SHA512 チェックサムと一致しない場合があります。

Cisco Nexus 3000、3100、および 3500 のコンパクトイメージ

モデル番号が **-XL** で終わらない Cisco Nexus 3000、3100、および 3500 シリーズ スイッチの初期モデルでは、ブートフラッシュに 1.4 ～ 1.6 ギガバイトのストレージスペースが割り当てられています。時間の経過とともに、NX-OS ソフトウェア イメージのファイルサイズは 1 ギガバイトを超えるまで少しずつ増加してきました。その結果、**-XL** で終了しないモデル番号の Nexus 3000、3100、および 3500 シリーズ スイッチでは、同時に複数の完全な NX-OS バイナリ イメージを保存できません。したがって、管理者は、**-XL** で終了しないモデル番号の Nexus 3000、3100、および 3500 シリーズ スイッチでは、Nexus 5000、6000、7000、および 9000 シリーズ スイッチなど、他の Nexus プラットフォームで使用される、NX-OS ソフトウェアの標準のアップグレード手順に従うことができません。

NX-OS ソフトウェア リリース 7.0(3)I3(3) 以降、NX-OS ソフトウェア イメージのファイルサイズは、NX-OS コンパクトイメージの手順を使用して縮小できます。これは、中断がないため、スイッチのコントロールプレーンやデータプレーン トラフィックの転送機能に影響を与えない手順です。

シスコのソフトウェア ダウンロード Web サイトのコンパクト NX-OS ソフトウェア イメージ

一部の NX-OS ソフトウェア リリースについては、[シスコのソフトウェア ダウンロード Web サイト](#) から、コンパクト NX-OS ソフトウェア イメージをダウンロードできます。これらのコンパクトイメージには、NX-OS バイナリ イメージファイルの整合性を検証するために使用できる公開された MD5/SHA512 チェックサムがあります。以下の NX-OS ソフトウェア リリース

については、[シスコのソフトウェアダウンロードWebサイト](#)から、コンパクトNX-OS ソフトウェア イメージをダウンロードできます。

- 9.3(4) 以降
- 9.2(4)
- 7.0 (3) I7 (8) 以降



(注) シスコの[ソフトウェアダウンロードWebサイト](#)で公開されているコンパクトNX-OS ソフトウェア イメージのMD5/SHA512 チェックサムは、NX-OS コンパクト イメージの手順で作成されたコンパクトイメージのMD5/SHA512 チェックサムと一致しない場合があります。

異なるスイッチ モデルで実行するコンパクト イメージ

NX-OS ソフトウェア リリース 9.3(6) を例にして、該当するスイッチの各モデルで実行する適切なコンパクト イメージを示します。

表 2: Cisco Nexus 3000、3100、および3500 シリーズプラットフォームそれぞれのコンパクトイメージテーブル

スイッチ モデル番号	コンパクト NX-OS ソフトウェア イメージのファイル名
N3K-C3016Q-40GE	n3000-compact.9.3.6.bin
N3K-C3048TP-1GE	n3000-compact.9.3.6.bin
N3K-C3064PQ-10GX	n3000-compact.9.3.6.bin
N3K-C3064TQ-10GT	n3000-compact.9.3.6.bin
N3K-C3064TQ-32T	n3000-compact.9.3.6.bin
N3K-C3132Q-40GE	n3100-compact.9.3.6.bin
N3K-C3132Q-40GX	n3100-compact.9.3.6.bin
N3K-C3172PQ-10GE	n3100-compact.9.3.6.bin
N3K-C3172TQ-10GT	n3100-compact.9.3.6.bin
N3K-C3172TQ-32T	n3100-compact.9.3.6.bin
N3K-C3524P-10G	n3500-compact.9.3.6.bin
N3K-C3524P-10GX	n3500-compact.9.3.6.bin
N3K-C3548P-10G	n3500-compact.9.3.6.bin
N3K-C3548P-10GX	n3500-compact.9.3.6.bin

NX-OS コンパクトイメージの手順

NX-OS コンパクトイメージの手順を開始するには、2つの方法があります。次のサブセクションでは、各オプションについて詳しく説明します。

ブートフラッシュ/USBでのNX-OS コンパクトイメージの手順

NX-OS ソフトウェアリリース7.0(3)I3(1)以降では、`install all` コマンドとコンパクトオプションを使用して、スイッチのブートフラッシュまたは接続された USB ドライブに保存されているイメージで、NX-OS コンパクトイメージ手順を開始できます。これは、スイッチが現在のブートで使用した NX-OS ソフトウェア イメージでも実行できます。ただし、現在ブート中のイメージを圧縮するには、スイッチのブートフラッシュに 750 MB 以上の空き領域が必要です。

スイッチのブートフラッシュに保存されている NX-OS ソフトウェア イメージで NX-OS コンパクトイメージの手順を開始する方法の例を次に示します。

```
switch# install all nxos bootflash:nxos.7.0.3.I7.8.bin compact
```

SCPによるNX-OS コンパクトイメージの手順

NX-OS ソフトウェア リリース 7.0(3)I5(2) 以降では、`copy` コマンドの追加のオプションを使用して、SCP（セキュアコピープロトコル）により、イメージをスイッチのブートフラッシュまたは接続された USB ドライブにコピーしながら、NX-OS コンパクトイメージの手順を開始できます。`copy` コマンドのコンパクトオプションは、イメージをスイッチのブートフラッシュまたは接続された USB ドライブに転送するときにイメージを圧縮するため、ブートフラッシュのスペースの制限を回避できます。このオプションは SCP でのみサポートされます。他のプロトコル（SFTP [セキュアファイル転送プロトコル] や FTP [ファイル転送プロトコル] など）はサポートされません。

NX-OS ソフトウェア イメージを SCP を介してスイッチのブートフラッシュにコピーする際に、NX-OS のコンパクトイメージの手順を開始する方法の例を次に示します。

```
switch# copy scp://username@192.0.2.100/nxos.7.0.3.I7.8.bin bootflash:nxos.7.0.3.I7.8.bin compact vrf management
```

NX-OS コンパクトイメージの手順シーケンス

NX-OS ソフトウェア イメージを圧縮する順序は重要です。スイッチのブートフラッシュサイズが制限されているため、ブートフラッシュに別の NX-OS ソフトウェア イメージが存在する場合、現在ロードされている NX-OS ソフトウェア イメージを圧縮することはできません。

まず、前述のブートフラッシュ/USBでのNX-OS コンパクトイメージの手順に従って、ブートフラッシュで現在ブートされているイメージを圧縮します。次に、可能であれば、前述の SCP による NX-OS コンパクトイメージ手順を使用して、目的の NX-OS ソフトウェア イメージをスイッチのブートフラッシュにコピーします。7.0(3)I5(2) よりも前の NX-OS ソフトウェア リリースにアップグレードするため、このオプションを使用できない場合は、目的の NX-OS ソフトウェア イメージをスイッチのブートフラッシュに通常どおりコピーしてから、NX-OS イメージを圧縮するための前述のブートフラッシュ/USBでの OS コンパクトイメージ手順に従います。

ブート中でないイメージを圧縮するには、スイッチのブートフラッシュに最低限 450 MB の空き領域が必要です。ターゲットイメージをコピーした後、スイッチのブートフラッシュで使用可能な空き領域が 450 MB 未満の場合は、接続した USB ドライブを使用するか、前述の SCP メソッドによる NX-OS コンパクトイメージの手順を使用してターゲットイメージを圧縮する必要があります。

NX-OS コンパクトイメージプラットフォームグループ

このドキュメントで説明する NX-OS コンパクトイメージの手順は、3 つの異なるプラットフォームグループに適用できます。

- Nexus 3000 デバイス (N3K-C3048、N3K-C3064 など)
- Nexus 3100 デバイス (N3K-C3132、N3K-C3172 など)
- Nexus 3500 デバイス (N3K-C3524、N3K-C3548 など)

圧縮された NX-OS バイナリ イメージファイルは、同じプラットフォームグループ内の他のデバイス間で再利用できます。ただし、コンパクト NX-OS バイナリ イメージファイルを、異なるプラットフォームグループに属するデバイスで使用することはできません。

たとえば、次のモデルの Nexus 3000 デバイスが 4 台あるシナリオを考えます。

- N3K-C3048TP-1GE
- N3K-C3064PQ-10GE
- N3K-C3172PQ-40GX
- N3K-C3548P-10G

N3K-C3048TP-1GE で圧縮された NX-OS バイナリ イメージファイルは、選択したファイル転送プロトコルを介して N3K-C3064PQ-10GE のブートフラッシュに直接転送できます

(N3K-C3064PQ-10GE に十分なスペースがある場合)。さらに、N3K-C3064PQ-10GE は、このコンパクト NX-OS バイナリ イメージファイルを使用して、サポートされている方法でアップグレードできます。ただし、同じコンパクト NX-OS バイナリ イメージファイルを使用して N3K-C3172PQ-40GX および N3K-C3548P-10G デバイスをアップグレードすることはできません。NX-OS コンパクトイメージの手順は、N3K-C3172PQ-40GX と N3K-C3548P-10G の両方のデバイスで個別に実行する必要があります。

同じプラットフォームグループ内の Nexus デバイス間のこの互換性を利用すれば、多数のデバイスの NX-OS ソフトウェア アップグレードを最適化できます。たとえば、100 台の N3K-C3048TP-1GE デバイスがある場合、単一のデバイスで NX-OS コンパクトイメージ手順を実施し、コンパクト NX-OS バイナリ イメージファイルを結果の 99 デバイスに転送できます。100 台すべてのデバイスで NX-OS コンパクトイメージの手順を実行する必要はありません。

スイッチを USB から起動する

ローダープロンプトを使用して USB からスイッチを起動するオプションが提供されています。

例：

```
loader> boot usb1:nxos.9.2.1.bin
```

BIOS およびパワー シーケンサ イメージのアップグレード

BIOS とパワー シーケンサが変更されることは稀ですが、変更があった場合は Cisco NX-OS イメージに組み込まれ、BIOS とパワーシーケンサがアップグレードされます。インストールプロセス中にインストーラから表示される概要に、BIOS とパワー シーケンサの現在のバージョンとターゲットのバージョンが含まれます。



(注) パワー シーケンサのアップグレードが正常に終了したら、システムの電源を切ってから入れ直す必要があります。

BIOS のアップグレード

BIOS のアップグレードについては、次の注意事項を参照してください。

Cisco Nexus 3000 シリーズ プラットフォームでは、リリース 7.0(3)I2(1) 以降のイメージをロードするために BIOS アップグレードが必要です。BIOS バージョン 3.xx のみが、すべての Cisco Nexus 3000 シリーズ スイッチのリリース 7.0(3)I2(1) 以降のリリースと互換性があります。これは通常の **install all** のアップグレード方法で処理されますが、高速リロードアップグレードの前に明示的にアップグレードする必要があります。

BIOS のアップグレードが完了した後でのみ、システムは新しいリリースへの高速リロードを実行できます。

各 Cisco Nexus 3000 シリーズ プラットフォームの BIOS バージョン

次の Cisco Nexus 3000 シリーズ プラットフォームの BIOS バージョンを参照してください。

表 3: Cisco Nexus 3000 シリーズ プラットフォームの BIOS バージョン テーブル

シリアル番号	スイッチ名	スイッチ モデル	最新の BIOS バージョン
1.	Cisco Nexus 3132Q スイッチ	C3132Q-40GE	5.2.0

シリアル番号	スイッチ名	スイッチ モデル	最新の BIOS バージョン
2.	Cisco Nexus 3132Q-X スイッチ	C3132Q-40GX	5.2.0
3.	Cisco Nexus 3172PQ スイッチ	C3172PQ-10GE	5.2.0
4.	Cisco Nexus 3172CR スイッチ	C3172PQ-10GE	5.2.0
5.	Cisco Nexus 3064-X スイッチ	C3064PQ-10GX	5.2.0
6.	Cisco Nexus 3064-TQ スイッチ	C3064TQ-10GT	5.0.0
7.	Cisco Nexus 3016Q スイッチ	C3016Q-40GE	5.0.0
8.	Cisco Nexus 3064-E スイッチ	C3064PQ-10GE	5.0.0
9.	Cisco Nexus 3064PQ スイッチ	C3064PQ-FA	5.0.0
10.	Cisco Nexus 3048TP スイッチ	C3048TP-1GE	5.0.0

高速リロードシナリオでのアップグレードの注意事項

手順

高速リロードを使用して新しいリリースにアップグレードします。Cisco NX-OS 9.x リリースへのアップグレードでサポートされている方法は、BIOS アップグレード後に `fast-reload` を使用するか、または `install all` を使用することだけです。次に例を示します。

```
switch#fast-reload nxos bootflash:nxos.9.2.1.bin
```

例：

- (注) 後でダウングレードするために構成が必要になるため、新しいリリースにアップグレードする前に構成をバックアップする必要があります。

非高速リロードシナリオでのアップグレードのガイドライン

高速リロードを使用していない場合に Cisco NX-OS リリース 9.x にアップグレードするには、所定の順序で次の手順を実行します。

手順

BIOS をアップグレードし、NX-OS イメージを Cisco NX-OS リリース 9.x にアップグレードするために **install all** を実行します。 **install all nxos bootflash:filename [no-reload | non-disruptive | non-interruptive | serial]** コマンドを使用して Cisco NX-OS ソフトウェアをアップグレードします。

例：

```
switch# install all nxos bootflash:nxos.9.2.1.bin
```

次のオプションを使用できます。

- **no-reload**：デバイスをリロードする前にソフトウェア アップグレードプロセスを終了します。
- **non-disruptive**：In-Service Software Upgrade (ISSU) を実行してデータトラフィックの中断を防ぎます。（デフォルトでは、ソフトウェア アップグレードプロセスは中断されません）。
- **non-interruptive**：プロンプトなしでソフトウェアをアップグレードします。このオプションは、エラーや健全性チェックをすべてスキップします。
- **シリアル**：システム内のラインカードを1つずつアップグレードします。（デフォルトでは、ラインカードは時間を節約するためにバッチでアップグレードされます）。

後でダウングレードするために構成が必要になるため、新しいリリースにアップグレードする前に構成をバックアップする必要があります。

アップグレードの前提条件

- アップグレードが進行している間は、ネットワークが安定しており変更が行われていないことを確認してください。
- 現在実行中のリリースとターゲットのリリースとの間の、機能の互換性も確認する必要があります。

アップグレード後の管理サービス

インバンドポートおよび管理ポートは、スイッチをアップグレード用にリセットする前に停止し、アップグレードの完了後に再開されます。この期間中、インバンドポートと管理ポートに対応したサービスが影響を受けます。

表 4: アップグレードのリセット中に影響を受けるインバンドポートおよび管理ポートのサービス

サービス	説明
[Telnet/SSH]	アップグレードがターゲットの Cisco NX-OS バージョンをロードするためにシステムをリセットすると、すべての Telnet/SSH セッションが切断されるため、アップグレードの完了後に再確立する必要があります。
AAA/RADIUS	AAA サービスを利用するアプリケーション（ログインなど）は、アップグレードプロセス中は無効になります。この期間中はすべてのネットワーク管理サービスが無効になるため、この動作が継続します。
HTTP	スイッチへの HTTP セッションはアップグレードのリポート中に切断されます。リポート後、HTTP は再起動され、スイッチが HTTP セッションを受け入れるようになります。
NTP	スイッチとの間の NTP セッションは、アップグレードのリポート中に中断されます。リポート後、保存されているスタートアップコンフィギュレーションに基づいて、NTP セッションが再確立されます。

レイヤ 2 プロトコルの影響

次の表に、レイヤ 2 プロトコルへのアップグレードの影響を示します。

表 5: レイヤ 2 プロトコルへのアップグレードの影響

プロトコル	説明
LACP	IEEE 802.3ad では、安定状態のときにデフォルトのスローエージングタイマーが 30 秒ごとに送信され、90 秒後に期限切れとなります。復旧期間は 90 秒未満のため、アップグレードは LACP に依存するピアに影響を与えないはずですが。
IGMP	IGMP はすでに存在するマルチキャストトラフィックの既存のフローを中断しませんが、新しいフローはアップグレードが完了するまで認識されません（ドロップされます）。この間、新しいルータポートまたはルータポートに対する変更は検出されません。

プロトコル	説明
DCBX および LLDP	DCBX は、LLDP を使用してピア デバイス間のパラメータを交換します。DCBX はリンク ローカルプロトコルであるため、スイッチがアップグレードを実施すると、アップグレードされているスイッチのすべてのポートで経過時間が増加します。 この間、手動による設定は無視されます。
CDP	アップグレード中は、存続可能時間の値がタイムアウトの推奨値よりも小さいと、増加されます（180 秒）。手動で指定されている設定は無視されます。
L2MP IS-IS	スイッチはアップグレード用にリブートする前に、L2 IS-IS hello をすべてのインターフェイスに送信し、ネイバー スイッチがアップグレード スイッチへのルートをダウンとしてマークしないようにします。この間のトポロジ変化も、アップグレードが完了するまで実行されません。

スイッチのイーサネット インターフェイス

コントロールプレーン停止期間中にリンク ダウンからリンク アップへの遷移を防止するため、動作上は停止しているが管理上は起動しているポートのレーザーが電源オフされます。この状況はアップグレードリブート開始状態時に発生します。アップグレードがリブートして再起動が正常に行われると、レーザーの電源が再びオンになります。この動作によって、リンク状態がアップグレード中にダウンからアップに遷移することが防止されます。

インストール前のチェック

健全性チェックを実行することにより、アップグレードに向けてシステムでの準備が完了していることを確認し、アップグレードの影響を理解しておく必要があります。

- ターゲットイメージが現在のイメージと機能上互換性があることを検証するために、**show incompatibility** コマンドを使用します。
- すべてのプロセスの重大度レベルが 5 以下に設定されているかどうか確認するために、**show logging level** コマンドを入力します。
- アップグレードの影響を識別するために、**show install all impact** コマンドを入力します。
- 最新の Cisco NX-OS ソフトウェアに更新するために、**install all** コマンドを入力します。
- インストーラの影響分析を確認し、次に進むかどうかを決定します



(注) スイッチはリロードされ、トラフィックが中断することがあります。

- インストールの進行状態をモニタします。
- アップグレードを検証します。
- インストールのステータスを検証するために、**show install all status** コマンドを入力します。

次の表に、ISSUの実行時に発生する可能性がある影響または潜在的な問題を確認する show コマンドを示します。

表 6: アップグレードの show コマンド

コマンド	定義
show incompatibility system	アップグレードバージョンに影響する、現在のシステムの非互換設定が表示されます。
show logging level	ファシリティ ロギング重大度設定を表示します。 アップグレードの実行時にすべてのプロセスのログ レベルが 5 以下に設定されていなければなりません。ログ レベルが 5 を超えるプロセスは、 show install all impact コマンドを入力しても表示されません。
show install all impact	アップグレードの影響を記述する情報を表示します。このコマンドは、アップグレードの中断の有無と、スイッチをリブートする必要があるかどうか、およびその理由についても表示します。

次の作業を実行して、問題が実際に発生する前に、発生する可能性のある問題を特定することもできます。

- **bootflash:** に、イメージを格納できるだけの空き容量があることを確認します。
- アップグレードバージョンに影響する、現在のシステムの非互換設定を表示します。

```
switch# show incompatibility system pcco.s
No incompatible configurations
```

- アップグレードの影響を確認します。

```
switch# show install all impact nxos bootflash:nxos.9.2.1.bin
```

高速リブートに関する情報

高速リブート中、CPU 上で実行されているイメージは新しいイメージをリロードし、CPU やファームウェアをリセットせずにこれを実行します。高速リブート中にトラフィックに短い中断がありますが、コールドリブート時よりも高速にスイッチがリロードできるようにします。

Cisco NX-OS ソフトウェアでは、非中断モードで高速リブートを使用できます。このモードでは、高速リブートはプロンプトなしでインストールプロセスを開始します。このリリースでは、高速リブートは、互換性のあるピアの BGP グレースフルリスタート (GR) もサポートします。 **trigger-gr** オプションを使用して、グレースフルリスタートで高速リブートをトリガーできます。

高速再起動のタイミング要件

高速再起動がサポートする構成のタイミング要件は次のとおりです。

- コントロールプレーンの中断後に ASIC をリセットし、データプレーンを中断するのにかかる時間 — コントロールプレーンが中断された場合、90 秒未満。
- トラフィックの転送を再開するのにかかる時間 — ASIC リセットから 30 秒未満。

高速リブートの注意事項

高速リブートは、限られた構成およびトポロジだけでサポートされます。サポートされている構成および注意事項の一部は、このセクションに記載されています。システムソフトウェアを **fast-reload** コマンドを使用して高速にリロードする際は、サポートされる次の注意事項を使用してください。

- 構成の変更 — リロードまたはアップグレード中は構成モードを開始できません。Cisco NX-OS ソフトウェアイメージをアップグレードまたはダウングレードする前に、アクティブなすべてのコンフィギュレーションセッションを保存、送信、または破棄する必要があります。アクティブなコンフィギュレーションセッションが、リロード中に警告なしで削除されます。

アクティブな構成セッションがないことを確認するには、**show configuration session summary** コマンドを使用します。

```
switch# show configuration session summary
There are no active configuration sessions
```

fast-reload コマンドを使用する前に、構成の互換性を確認してください。



- (注) カーネルまたは BIOS の変更につながる可能性のあるアップグレードには、**fast-reload** コマンドを使用しないでください。

構成セッションの詳細については、『Cisco Nexus 3000 シリーズ NX-OS システム管理構成ガイド』を参照してください。



(注) CLI および SNMP の構成変更の要求は拒否されます。

- トポロジ—アップグレードを実行する前に Spanning Tree Protocol (STP) などのトポロジ変更を行う必要があります。モジュールのインストールまたは削除を行うのは、アップグレードの前または後だけにしてください。ただし、レイヤ2およびルーティングトポロジは変更しないでください。また、デフォルトルートブリッジは設定しないでください。
- スケジューリング：アップグレードは、ネットワークが安定しているときにスケジュールしてください。スイッチまたはネットワークにアクセスできるすべての人がその時間にスイッチまたはネットワークを設定しないようにします。アップグレード中はスイッチを設定できません。
- スペース：イメージをコピーする場所に十分なスペースがあることを確認します。内部ブートフラッシュにはおよそ 650 MB の空き容量が必要です。
- ハードウェア：インストールプロセスの最中は、電源停止が起これないようにします。電源が停止すると、ソフトウェアイメージが破損するおそれがあります。
- リモートサーバへの接続：10/100/1000 BASE-T イーサネットポート接続の IPv4 アドレスまたは IPv6 アドレスを設定します（インターフェイス mgmt0）。スイッチにリモートサーバへのルートが設定されているかどうかを確認します。サブネット間でトラフィックをルーティングするルータがない場合は、スイッチおよびリモートサーバは同一のサブネットワーク上にある必要があります。
- リンク集約制御プロトコル (LACP) 高速タイマー：高速リロードするには、LACP 高速タイマーを設定しないでください。
- 互換性イメージは、次の 2 つの方法のいずれかで取得します。
 - ローカル—スイッチ上のローカルイメージを使用できます。
 - リモート—イメージがリモートの場所にある場合、リモートサーバのパラメータとローカルで使用するファイル名を使用して、取得先を指定します。
- コマンド：新しいソフトウェアの準備およびインストールを行うには、次のコマンドを使用します。
 - リモートサーバへの接続を確認するには **ping** コマンドを使用します。
 - イメージファイルをコピーするために必要なスペースがあるかどうかを確認するには **dir** コマンドを使用します。
 - アップグレードの影響を識別するために、**show install all impact** コマンドを入力します。また、アップグレードが中断を伴うかどうか、あるいはアップグレードが中断する理由と、スイッチをリブートする必要があるかどうかや、リブートが必要な理由も表示されます。



(注) コンソールポートにログインして、アップグレードプロセスを開始することを推奨します。

- コントロールプレーンの中断とデータプレーンの中断の間は、CPUの応答が停止します。
- 構成 — 高速リブートは現在次の設定をサポートします。
 - BGP v4 および v6
 - 16 方向 ECMP
 - 48 個のダウンリンク L2 ポート
 - 4 個の SVI
 - 10 個以下の VLAN 未満
 - 2000 v4 ルートおよび 2000 v6 ルート
 - RACL
 - ARP
 - STP エッジポート構成
- リペイブされたリロードは、ポートモードプロファイル、URPFの有効化/無効化、TCAMの再カービングなど、以前は完全なリロードが必要だった設定の変更もサポートします。
- EOLに達したバージョンとの間のアップグレードには、**fast-reload** オプションを使用しないでください。
- 高速リロード機能は、ポートモードプロファイル、URPFの有効化/無効化、TCAMの再カービングなど、以前は完全なリロードが必要だった設定の変更もサポートします。
- 高速リロードは BIOS をアップグレードしません。
- CoPP 機能をサポートする Cisco NX-OS リリースに CoPP 機能をサポートしないリリースからアップグレードする場合は、アップグレード後にデバイスの CoPP を有効にするためにセットアップユーティリティを実行する必要があります。
- 高速リロードのアップグレードは、Cisco NX-OS リリース 7.0(3)I7(4) または 7.0(3)I7(5) から Cisco NX-OS 9.2(x) リリースでサポートされます。
- Cisco NX-OS リリース 9.3(10) 以降、高速リロードのサポートは NX-OS CLI で「非表示」になっていますが、コマンドを実行した場合は引き続き使用できます。高速リロード機能では、今後のリリース 9.3(x) トレインは進められません。
- シスコは、Cisco Nexus 3164Q スイッチの高速リロードパフォーマンスが 30 秒未満であると主張する公開ドキュメントをすべて削除します。

- **install-all** コマンドを使用して Cisco NXOS リリース 6.0(2)U2(1) より前のリリースにダウングレードすると、高速リロードは機能しません。

Fast-Reload コマンドの使用

fast-reload コマンドは Cisco Nexus 3000 または Nexus 3100 シリーズ スイッチを **reload** コマンドよりも高速にリロードします。

ASCII 構成ベースの高速リロードもまた、PSS/バイナリ構成ベースの高速リロードに加えてサポートされます。Cisco Nexus 3000 シリーズ スイッチは、高速リロードでは完全にリセットされません。Cisco Nexus 3000 および 3100 シリーズ スイッチのボードには、2つのリセットドメインがあります。リセットドメイン1には、CPU、プラットフォーム コントローラ ハブ (PCH)、管理イーサネット コントローラ、PCIブリッジ、OBFL、USB、およびファン コントローラが含まれます。リセットドメイン2には、ASIC、PHYリタイマー、SFP+モジュール、およびQSFPモジュールが含まれています。高速リロードはリセットドメイン2のみをリセットし、CPUのリセット、BIOSのリロード、およびファームウェアはありません。

fast-reload コマンドを実行すると、次の一連のイベントが発生します。

1. スイッチは、NXOS ソフトウェア イメージをロードし、カーネルをアップグレードします。すべてのアプリケーションはステートレス コールドリブートされ、スタートアップ コンフィギュレーションを介して再起動します。
2. コントロールプレーンが中断されます。コントロールプレーンの中断時に、すべての制御プロトコル通信が停止します。コントロールプレーンの中断は常に 90 秒未満です。
3. コントロールプレーンの中断後にすべてのコントロールプレーンのアプリケーションはステートレス コールドリブートされ、状態は保持されません。新しい設定は、スイッチのリロード時に適用されます。
4. データプレーンが中断されます。データプレーンの中断は常に 30 秒未満です。
5. フォワーディングプレーンでは、すべてのリンクが使用できなくなり、データプレーンの状態はリロード後は保持されません。トラフィックの転送が 30 秒以内に再開します。



(注) 作業イメージがあることを確認し、このコマンドを使用する前に高速リブート操作の影響を分析してください。

PSS/バイナリ構成での高速リロード

手順

ステップ1 スイッチにログインします。

ステップ 2 Cisco Nexus 3000 および 3100 シリーズ スイッチで高速リブートを実行するには、次のコマンドを使用します。 **fast-reload [save-config] [trigger-gr] [nxos bootflash:nxos-image-name] [non-interruptive]** コマンドを使用して、高速リロードを実行します。

ASCII 構成での高速リロード

PSS/バイナリ構成ベースの高速リロードに加えて、ASCII 構成ベースの高速リロードもサポートされています。



- (注) コピー ファイルの起動と高速リロードは、特定の設定、つまり、ポート チャネルを備えたレイヤ 3 ポート、eBGP、およびホストへの SVI のみを持ついくつかの物理 I2 ポートでのみサポートされます。

ASCII ファイル ベースの高速リロードを使用するには、次のコマンドを使用します。

copy configuration-file startup-config

fast-reload nxos bootflash:nxos-image-name

構成ファイルは、高速リロードがアップグレードまたは高速リロードで使用するシステム構成を含む ASCII ファイルです。リモートロケーションからもコピーできます。NXOS ソフトウェアイメージが指定されていない場合、スイッチに存在するイメージがリロードされます。提供された NXOS ソフトウェア イメージが既存のバージョンよりも高いバージョンである場合、アップグレードがトリガーされます。



- (注) 後続の高速再起動操作が NXOS ソフトウェア イメージをブート変数として使用するには、**fast-reload** コマンドを実行する際に **save-config** オプションを指定します。**save-config** オプションが指定されていない場合、**fast-reload** コマンドはブート変数を保存しません。

高速リロードのアップグレード

install all コマンドと比較して、データのダウンタイムを最小限に抑えながら、より高速なリロードと新しいリリースへのアップグレードを実行できます。

この機能は、リリース 6.0(2)U2(1) 以降の Cisco Nexus 3000 シリーズ スイッチでの既存の高速リロードサポートに似ていますが、リリース 7.0(3)I2(1) 以降のリリースにアップグレードする前に BIOS をアップグレードするためのいくつかの追加手順があります。



注意 リリース 7.0(3)I2(1) への高速リロードは、BIOS がアップグレードされた後のリリース 6.0(2)U6(3a) 以降のリリースでのみサポートされます。BIOS をアップグレードせずにリリース 6.0(2)U6(3a) からリリース 7.0(3)I2(1) に高速リロードを使用すると、スイッチが起動しません。詳細については、BIOS アップグレードを参照してください。



(注) ゲスト シェルまたは仮想サービスが有効になっている場合は、**install all** コマンドを使用する必要があります。**show virtual-service list** コマンドを使用して、ゲストシェルまたは仮想サービスの存在または状態を確認します。

高速リブートによる BGP グレースフル リスタートを有効にする

Cisco NX-OS ソフトウェアを使用すると、高速リブートによる BGP グレースフル リスタート (GR) を有効にすることができます。**fast-reload trigger-gr** コマンドを使用して、BGP GR を有効にできるようになりました。このコマンドは、すべての BGP ピアが GR 対応の場合にのみ使用してください。

Cisco Nexus 3000 および 3100 シリーズ スイッチで高速リブートで BGP GR を有効にするには、次のコマンドを使用します。

fast-reload [save-config] [trigger-gr] [nxos bootflash:nxos-image-name] [non-interruptive] コマンドを使用して、高速リロードを実行します。

```
switch# fast-reload trigger-gr nxos bootflash:nxos.9.2.1.bin non-interruptive
```

高速リブートを使用したアップグレードとダウングレード

高速リブートを使用したアップグレード

高速リブートを使用してスイッチでソフトウェアをアップグレードできます。アップグレードするには、NXOS ソフトウェア イメージを指定する必要があります。

はじめる前に

fast-reload コマンドで指定されている NXOS ソフトウェア イメージのバージョンが現在スイッチにあるイメージのバージョンよりも上位のものであることを確認します。

```
switch# fast-reload nxos bootflash:nxos.9.2.1.bin
```

高速リブートを使用したダウングレード

高速リブートを使用したシステムソフトウェアのダウングレードはサポートされていません。イメージソフトウェアをダウングレードするには、高速リブートの代わりに **install all** コマンドを使用します。

install all コマンドの使用

install all コマンドは、Cisco Nexus 3000 および Nexus 3100 シリーズ スイッチで中断を伴うソフトウェアのインストールをトリガーします。次のイメージがインストール中にアップグレードされます。

- NX-OS ソフトウェア イメージ
- システム BIOS
- システムのパワー シーケンサ

install all コマンドには、次の利点があります。

- 1 つのコマンドのみを使用して Cisco Nexus 3000 シリーズ スイッチをアップグレードできます。
- インストールを続行する前に、システムに加えようとしている変更についての説明を参照できます。たとえば、アップグレードで中断が生じる可能性があるかどうかを確認されます。
- 次の質問が表示された場合、アップグレードを続行するか、または取り消すことができます（デフォルトは **no** です）。

```
Do you want to continue (y/n) [n]: y
```
- **install all non-interruptive** コマンドを使用して、プロンプトなしで新しいイメージをインストールすることもできます。
- このコマンドは、イメージの整合性を自動的に確認します。これには、NXOS ソフトウェア イメージが含まれます。
- プラットフォームの有効性チェックが実行され、正しくないイメージが使用されていないかどうかを確認されます。
- **Ctrl+C** を押すと、**install all** コマンドを正常に終了できます。このコマンドシーケンスは、進行中の更新ステップを完了してから、EXEC プロンプトに戻ります。
- **install all** コマンドの入力後、シーケンス内のいずれかのステップで問題が発生した場合、アップグレードは終了します。
- パワーシーケンサをアップグレードした場合の影響について警告するメッセージが、次のように表示されます。

```
Warning: please do not remove or power off the module at this time.
```

```
Note: Power-seq upgrade needs a power-cycle to take into effect.
```



- (注) パワー シーケンサのアップグレードが正常に終了したら、システムの電源を切ってから入れ直す必要があります。

Install All Non-Interruptive コマンドの使用

Cisco NX-OS ソフトウェアは、中断のない **install all** コマンドの使用をサポートしています。**install all non-interruptive** コマンドを使用して、プロンプトなしで新しいイメージをインストールできるようになりました。

Cisco Nexus 3000 および 3100 シリーズ スイッチで中断のないインストールを実行するには、次のコマンドを使用します。

install all nxos bootflash: [nxos-image-name] [non-interruptive]

```
switch# install all nxos bootflash:nxos.9.2.1.bin
```

アップグレード手順

アップグレードプロセスは、**install all** コマンドを入力すると起動されます。ここでは、単一の Cisco Nexus 3000 シリーズ スイッチをアップグレードする際に発生するイベントのシーケンスについて説明します。



- (注) リリース 7.0(3)I2(1) より前のリリースを使用している場合は、最初に Cisco Nexus 3000 リリース 6.0.2.U6(3a) にアップグレードしてから、リリース 7.0(3)I2(1) 以降のリリースにアップグレードします。



- (注) 互換性チェック中、Reason フィールドに次の ISSU 関連のメッセージが表示されることがあります。

表 7: ISSU 関連のメッセージ

Reason フィールドメッセージ: Cisco NX-OS リリース 7.0(3)I3(1)	Reason フィールドメッセージ: Cisco NX-OS リリース 7.0(3)I4(1) 以降のリリース	説明
Incompatible image	ISSU の非互換イメージ	アップグレードしようとする Cisco NX-OS イメージは ISSU をサポートしていません。

Reason フィールド メッセージ : Cisco NX-OS リリース 7.0(3)I3(1)	Reason フィールド メッセージ : Cisco NX-OS リリース 7.0(3)I4(1) 以降のリリース	説明
Hitless upgrade is not supported	デフォルト アップグレードはヒットレスではありません	デフォルトでは、ソフトウェアアップグレードプロセスは中断されます。ISSUを実行するには、non-disruptive オプションを設定する必要があります。

インストールの概要

次の表に、アップグレードプロセスの概要を示します。

表 8: アップグレード プロセスの概要

アップグレードの準備	
------------	--

1. 最初の Cisco Nexus 3000 シリーズ スイッチにログインします。コンソールポートにログインすることを推奨します。vPC トポロジでは、トポロジ内のプライマリスイッチまたはセカンダリスイッチで最初のアップグレードが実行できます。
2. Cisco.com にログインし、Software Download Center にアクセスします。Cisco.com にログインするには、<https://www.cisco.com> に進み、ページ最上部の[ログイン]をクリックします。シスコユーザー名およびパスワードを入力してください。
3. ソフトウェアイメージを選択してサーバーにダウンロードします。
4. イメージファイルをコピーするために必要なスペースが bootflash: ディレクトリにあることを確認します。
5. bootflash: ディレクトリにさらに多くのスペースが必要な場合、不要なファイルを削除して使用できるスペースを作ります。
6. ftp:、http:、https:、tftp:、scp:、sftp などの転送プロトコルを使用して、CiscoNX-OS ソフトウェアイメージをブートフラッシュにコピーします。例：

```
switch# copy
scp://user@scpserver.cisco.com//download/nxos.9.3.1.bin
bootflash:nxos.9.3.1.bin
```

コピー実行時にソフトウェアイメージサイズを縮小するには、**compact** キーワードを使用します。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.
```

(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。

7. **dir bootflash** コマンドを使用して、転送されたイメージのサイズを比較します。<https://www.cisco.com> から取得したイメージのファイルサイズと、転送されたファイルのイメージサイズが同じになっている

	<p>る必要があります。</p> <p>8. トポロジ内の各スイッチについて、次の手順を実行します。</p>
アップグレード前のチェック	<p>1. ターゲットイメージが現在のイメージと機能上互換性があることを検証するために、show incompatibility コマンドを使用します。</p> <p>2. アップグレードの影響を識別するために、show install all impact コマンドを入力します。</p>
アップグレードの開始	<p>1. 最新の Cisco NX-OS ソフトウェアに更新するために、install all コマンドを入力します。</p> <p>2. インストーラの影響分析を確認し、次に進みます。</p> <p>3. インストーラーはソフトウェアをアップグレードします。</p>
アップグレードの検証	<p>1. インストールのステータスを検証するために、show install all status コマンドを入力します。</p>

外部フラッシュメモリ デバイスからの実行構成のコピー

外部フラッシュメモリ デバイスからコンフィギュレーション ファイルをコピーできます。

始める前に

外部フラッシュメモリ デバイスを、アクティブなスーパーバイザ モジュールに挿入します。

手順

	コマンドまたはアクション	目的
ステップ 1	<p>(任意) dir {usb1: usb2:}[directory/]</p> <p>例 :</p> <pre>switch# dir usb1:</pre>	外部フラッシュメモリ デバイス上のファイルを表示します。
ステップ 2	<p>copy {usb1: usb2:}[directory/] filename {bootflash:}[directory/] filename</p> <p>例 :</p> <pre>switch# copy usb1:pcco.k bootflash:pcco.k</pre>	外部フラッシュメモリ デバイスからブートフラッシュにイメージをコピーします。filename の引数では大文字と小文字が区別されます。
ステップ 3	<p>copy {usb1: usb2:}[directory/] filename running-config</p> <p>例 :</p> <pre>switch# copy usb1:dsn-config.cfg running-config</pre>	外部フラッシュメモリ デバイスから実行コンフィギュレーションをコピーします。filename の引数では大文字と小文字が区別されます。

	コマンドまたはアクション	目的
ステップ 4	(任意) show running-config 例： switch# show running-config	実行設定を表示します。
ステップ 5	(任意) copy running-config startup-config 例： switch# copy running-config startup-config	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

外部フラッシュメモリ デバイスからのスタートアップ構成のコピー

Cisco NX-OS デバイス上のスタートアップ コンフィギュレーションを復元するには、外部フラッシュメモリ デバイスに保存された新しいスタートアップ コンフィギュレーション ファイルをダウンロードします。

始める前に

外部フラッシュメモリ デバイスを、アクティブなスーパーバイザ モジュールに挿入します。

手順

	コマンドまたはアクション	目的
ステップ 1	(任意) dir {usb1: usb2:}[directory/] 例： switch# dir usb1:	外部フラッシュメモリ デバイス上のファイルを表示します。
ステップ 2	copy {usb1: usb2:}[directory/] filename {bootflash:}[directory/]filename 例： switch# copy usb1:pcco.k bootflash:pcco.k.	外部フラッシュメモリ デバイスからブートフラッシュにイメージをコピーします。filename の引数では大文字と小文字が区別されます。
ステップ 3	copy {usb1: usb2:}[directory/] filename startup-config 例： switch# copy usb1:dsn-config.cfg startup-config	外部フラッシュメモリ デバイスからスタートアップ コンフィギュレーションをコピーします。filename の引数では大文字と小文字が区別されます。
ステップ 4	(任意) show startup-config 例： switch# show startup-config	スタートアップ コンフィギュレーションを表示します。

	コマンドまたはアクション	目的
ステップ 5	(任意) copy running-config startup-config 例 : <pre>switch# copy running-config startup-config</pre>	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

vPC 以外のトポロジでのアップグレードプロセス

次のリストは、vPC 以外のトポロジでのアップグレードプロセスを要約したものです。

1. **install all** コマンドは、インストールのアップグレードをトリガーします。
2. 互換性チェックにより、アップグレードの影響が表示されます。
3. アップグレードの影響に基づき、インストールが続行するか、停止します。
4. 現在の状態が保存されます。
5. システムが新しいイメージをアンロードして実行します。
6. システム ソフトウェアとアプリケーションの再起動が正常に行われます。
7. 新しいイメージでインストーラが再開されます。
8. インストールが完了します。

次の例では、アップグレードプロセスを表示します。

```
switch# install all nxos bootflash:<nxos-image-name>
```

vPC のアップグレードプロセス

プライマリスイッチでの vPC トポロジのアップグレードプロセス

次のリストは、プライマリまたはオペレーショナルプライマリ vPC ロールのいずれかを有する vPC トポロジのスイッチ上のアップグレードプロセスを要約したものです。vPC 以外のトポロジでのスイッチのアップグレードと異なるステップは太字で示しています。



(注) vPC トポロジでは、2 台のピアスイッチを個別にアップグレードする必要があります。一方のピアスイッチでアップグレードを実行しても、vPC ピアスイッチは自動的に更新されません。

1. vPC プライマリスイッチで **install all** コマンドを発行すると、インストールのアップグレードが開始されます。

2. 互換性チェックにより、アップグレードの影響が表示されます。
3. アップグレードの影響に基づき、インストールが継続するか、停止します。
4. 両方の vPC ピア スイッチで設定がロックされます。
5. 現在の状態が保存されます。
6. システムが新しいイメージをアンロードして実行します。
7. システム ソフトウェアとアプリケーションの再起動が正常に行われます。
8. 新しいイメージでインストーラが再開されます。
9. これでインストールが完了しました。

インストールが完了すると、vPC プライマリ スイッチがアップグレードされます。



- (注) vPC プライマリ スイッチがアップグレードされたバージョンを実行し、vPC セカンダリ スイッチが元のバージョンのソフトウェアを実行します。

セカンダリ スイッチでの vPC トポロジのアップグレード プロセス

次のリストは、セカンダリまたはオペレーショナルセカンダリ vPC ロールのいずれかを有する vPC トポロジのスイッチ上のアップグレードプロセスを要約したものです。vPC 以外のトポロジでのスイッチのアップグレードと異なるステップは太字で示しています。

1. **vPC セカンダリ スイッチで `install all` コマンドを発行すると、インストールのアップグレードが開始されます。**
2. 互換性チェックにより、アップグレードの影響が表示されます。
3. アップグレードの影響に基づき、インストールが継続するか、停止します。
4. 現在の状態が保存されます。
5. システムが新しいイメージをアンロードして実行します。
6. システム ソフトウェアとアプリケーションの再起動が正常に行われます。
7. 新しいイメージでインストーラが再開されます。
8. **プライマリ スイッチとセカンダリ スイッチでの設定のロックが解除されます。**
9. これでインストールが完了しました。

Nexus 9000 -R シリーズスイッチのvPCアップグレードおよびダウングレード手順

vPC トポロジでは、通常 2 台のピアスイッチを個別にアップグレードする必要があります。一方のピアスイッチでアップグレードを実行しても、vPC ピアスイッチは自動的に更新されません。

ただし、Cisco NX-OS リリース 7.0(3)F3(3c) および 7.0(3)F3(4) は、vPC ピアスイッチの Cisco NX-OS リリース 9.2(x) とは互換性がありません。両方の vPC ピアを同時に Cisco NX-OS リリース 9.2(x) にアップグレードして、1 つのスイッチが 7.0(3)F3(x) リリースを実行し、もう一方のスイッチが 9.2(x) を実行している状況を避ける必要があります。スイッチを Cisco NX-OS リリース 7.0(3)F3(4) からアップグレードする場合は、次の手順を使用してアップグレード中のトラフィックの影響を最小限に抑えることも可能です。



(注) この手順は、Broadcom または Cloudscale ベースのスイッチでは使用できません。

1. スイッチ A と B は、Cisco NX-OS リリースを実行しています。スイッチ A は、プライマリスイッチで、スイッチ B はセカンダリスイッチです。両方のスイッチで **copy r s** コマンドを使用します。

```
primary_switch# show vpc role
vPC Role status
-----
vPC role : primary
vPC system-mac : 00:23:04:ee:be:64
vPC system-priority : 32667
vPC local system-mac : 70:df:2f:eb:86:1f
vPC local role-priority : 90
vPC peer system-mac : 70:df:2f:eb:1c:ab
vPC peer role-priority : 100
primary_switch#

secondary_switch# show vpc role
vPC Role status
-----
vPC role : secondary
vPC system-mac : 00:23:04:ee:be:64
vPC system-priority : 32667
vPC local system-mac : 70:df:2f:eb:1c:ab
vPC local role-priority : 100
vPC peer system-mac : 70:df:2f:eb:86:1f
vPC peer role-priority : 90
secondary_switch#

primary_switch# copy r s v
[#####] 100%
Copy complete.

secondary_switch# copy r s v
[#####] 100%
Copy complete.
```

2. プライマリスイッチでピアリンク (PL) を停止します。セカンダリスイッチは、そのvPC レッグを停止します。

```

primary_switch# conf t
Enter configuration commands, one per line. End with CNTL/Z.
primary_switch(config)# int port-channel 100
primary_switch(config-if)# shutdown

Reload the secondary switch with Release 9.2.1 image (change bootvar /reload)

secondary_switch(config)# boot nxos nxos.9.2.1.bin
Performing image verification and compatibility check, please wait....
secondary_switch(config)#
secondary_switch(config)# copy r s v
[#####] 100%
Copy complete.

secondary_switch# reload
This command will reboot the system. (y/n)? [n] y

After reload
-----
secondary_switch# show vpc
Legend:
(*) - local vPC is down, forwarding via vPC peer-link
vPC domain id : 100
Peer status : peer link is down
vPC keep-alive status : peer is alive
Configuration consistency status : failed
Per-vlan consistency status : success
Configuration inconsistency reason: Consistency Check Not Performed
Type-2 inconsistency reason : Consistency Check Not Performed
vPC role : none established
Number of vPCs configured : 20
Peer Gateway : Enabled
Dual-active excluded VLANs : -
Graceful Consistency Check : Disabled (due to peer configuration)
Auto-recovery status : Disabled
Delay-restore status : Timer is off.(timeout = 90s)
Delay-restore SVI status : Timer is off.(timeout = 10s)
Operational Layer3 Peer-router : Disabled
vPC Peer-link status
-----
id Port Status Active vlans
-----
1 Po100 down -

secondary_switch#

primary_switch(config-if)# show vpc
Legend:
(*) - local vPC is down, forwarding via vPC peer-link
vPC domain id : 100
Peer status : peer link is down
vPC keep-alive status : peer is alive
Configuration consistency status : success
Per-vlan consistency status : success
Type-2 consistency status : success
vPC role : primary
Number of vPCs configured : 20
Peer Gateway : Enabled

```

```

Peer gateway excluded VLANs : -
Dual-active excluded VLANs and BDs : -
Graceful Consistency Check : Enabled
Auto-recovery status : Enabled, timer is off.(timeout = 240s)
Operational Layer3 Peer-router : Disabled
vPC Peer-link status
-----
id Port Status Active vlans
-- --
1 Po100 down -

```

3. セカンダリ スイッチ上の vPC ドメインで、vPC 自動リカバリを設定します。 **vpc upgrade** を有効化します (exec コマンド)。

```

secondary_switch(config)# vpc domain 100
secondary_switch(config-vpc-domain)# auto-recovery
secondary_switch(config-vpc-domain)# end

secondary_switch# show running-config vpc
!Command: show running-config vpc
!Running configuration last done at: Wed May 16 06:34:10 2018
!Time: Wed May 16 06:34:14 2018
version 9.2(1) Bios:version 01.11
feature vpc
vpc domain 100
peer-switch
role priority 100
peer-keepalive destination 10.1.31.30 source 10.1.31.29
delay restore 90
peer-gateway
auto-recovery
ipv6 nd synchronize
ip arp synchronize
interface port-channel100
vpc peer-link
interface port-channel2001
vpc 101

secondary_switch# show vpc upgrade
vPC upgrade : TRUE
SVI Timer : 0
Delay Restore Timer : 0
Delay Orphan Port Timer : 0
secondary_switch#

secondary_switch# show vpc upgrade >> Hidden command
vPC upgrade : FALSE
SVI Timer : 10
Delay Restore Timer : 90
Delay Orphan Port Timer : 0

secondary_switch# vpc upgrade >> Hidden command

```

4. セカンダリ スイッチでレイヤ 3 のルートが学習された後に、新しいリリース イメージでプライマリ スイッチをリロードします。セカンダリ スイッチがプライマリの役割を引き継ぎ、約 5 秒後にその vPC レッグを稼働します。


```

primary_switch(config)# show boot
Current Boot Variables:
sup-1
NXOS variable = bootflash:/nxos.9.2.1.bin
No module boot variable set
Boot Variables on next reload:
sup-1
NXOS variable = bootflash:/nxos.9.2.1.bin

No module boot variable set
primary_switch(config)# end

primary_switch# show boot
Current Boot Variables:
sup-1
NXOS variable = bootflash:/nxos.9.2.1.bin
No module boot variable set
Boot Variables on next reload:
sup-1
NXOS variable = bootflash:/nxos.9.2.1.bin

No module boot variable set
primary_switch# reload
This command will reboot the system. (y/n)? [n] y

secondary_switch# show vpc
Legend:
(*) - local vPC is down, forwarding via vPC peer-link
vPC domain id : 100
Peer status : peer link is down
vPC keep-alive status : peer is not reachable through peer-keepalive
Configuration consistency status : failed
Per-vlan consistency status : success
Configuration inconsistency reason: Consistency Check Not Performed
Type-2 inconsistency reason : Consistency Check Not Performed
vPC role : primary
Number of vPCs configured : 20
Peer Gateway : Enabled
Dual-active excluded VLANs : -
Graceful Consistency Check : Disabled (due to peer configuration)
Auto-recovery status : Enabled, timer is off.(timeout = 240s)
Delay-restore status : Timer is off.(timeout = 0s)
Delay-restore SVI status : Timer is off.(timeout = 0s)
Operational Layer3 Peer-router : Disabled
vPC Peer-link status
-----
id Port Status Active vlans
-- ----
1 Po100 down -
vPC status

```

5. プライマリ スイッチが復帰したら、そのピア リンクが稼働します。

```

primary_switch# show vpc
Legend:
(*) - local vPC is down, forwarding via vPC peer-link
vPC domain id : 100
Peer status : peer adjacency formed ok
vPC keep-alive status : peer is alive
Configuration consistency status : success
Per-vlan consistency status : success

```

```

Type-2 consistency status : success
vPC role : primary, operational secondary
Number of vPCs configured : 20
Peer Gateway : Enabled
Dual-active excluded VLANs : -
Graceful Consistency Check : Enabled
Auto-recovery status : Disabled
Delay-restore status : Timer is off.(timeout = 90s)
Delay-restore SVI status : Timer is off.(timeout = 10s)
Operational Layer3 Peer-router : Disabled
vPC Peer-link status
-----
id Port Status Active vlans
-----
1 Po100 up 1,101-400

```

ダウングレードの場合は、両方のスイッチを同時にリロードします。

アップグレードステータスのモニタリング

次の表に、インストールのアップグレードのモニタリングに使用される **show** コマンドを示します。

コマンド	定義
show install all failure-reason	インストール時に失敗したアプリケーションと、インストールが失敗した理由が表示されます。
show install all status	インストールの高レベル ログが表示されます。
show tech-support	問題を報告するときに、Cisco Technical Assistance Center に提供可能なシステム情報およびコンフィギュレーション情報が表示されます。

NX-OS アップグレード履歴

Cisco Nexus 3000 スイッチの耐用期間中は、多くのアップグレード手順を実行できます。メンテナンス目的で実行する場合や、オペレーティングシステムを更新して新しい機能を取得する場合にアップグレードが発生します。時間の経過とともに、スイッチは何度も更新される可能性があります。アップグレードのタイプと発生時刻を表示すると、問題のトラブルシューティングやスイッチの履歴の把握に役立ちます。

Cisco NX-OS リリース 9.3(5) 以降、Cisco Nexus 3000 スイッチは、時間の経過とともに実行されたすべてのアップグレードアクティビティをログに記録し、これらのイベントの包括的な履歴を提供します。保存されているアップグレード履歴タイプは次のとおりです。

- Cisco NX-OS システムのアップグレード
- 9Electronic Programmable Logic Device (EPLD) アップグレード

- Software Maintenance Upgrade (SMU) インストール

show upgrade history コマンドを入力して、Cisco NX-OS のアップグレード履歴を表示します。出力には、以前にスイッチで発生したアップグレードアクティビティが表示され、各イベントの開始時刻と終了時刻が定義されます。**show upgrade history** コマンドの出力例を次に示します。

```
switch# show upgrade history
TYPE                VERSION  DATE                STATUS
NXOS EPLD           n3000-   26 Apr 2020 11:37:16  EPLD Upgrade completed
                   epld.9.3.4.img
NXOS EPLD           n3000-   26 Apr 2020 11:32:41  EPLD Upgrade started
                   epld.9.3.4.img
NXOS system image  9.3(5)   24 Mar 2020 20:09:10  Installation End
NXOS system image  9.3(5)   24 Mar 2020 20:05:29  Installation started
NXOS SMU            9.3(5)   03 Mar 2020 23:34:15  Patch activation ended for
                   nxos.libnbproxyccli_patch-n3k_
                   ALL-1.0.0-9.3.5.lib32_n3000.rpm
NXOS SMU            9.3(5)   03 Mar 2020 23:34:03  Patch activation started for
                   nxos.libnbproxyccli_patch-n3k_
                   ALL-1.0.0-9.3.5.lib32_n3000.rpm
```

以前のリリースへのダウングレード

スイッチをダウングレードするために **install all** コマンドを入力する手順はスイッチをアップグレードするために **install all** コマンドを使用するのと同じですが、ロードするイメージファイルは、スイッチで現在実行しているイメージよりも前のリリースになります。**show incompatibility system command** コマンドを使用すると、現在のリリースとターゲットリリースとの間に機能の非互換性がないことを確認できます。ダウングレードは中断を伴うことに注意してください。



- (注) 特定リリースにダウングレードする前に、スイッチにインストールされている現在のリリースのリリースノートを確認し、ハードウェアにその特定リリースとの互換性があることを確認してください。詳細については、『[Nexus 3000 Series Switch Release Notes](#)』を参照してください。



- (注) 次の制限は、Trust Anchor Module (TAM) をサポートする Cisco Nexus プラットフォームスイッチに適用されます。

Cisco NX-OS リリース 9.3(3) 以降から以前のバージョンにダウングレードする場合、TACACS グローバル キーは復元できません。TAM は 9.3(3) でバージョン 7 に更新されましたが、以前の NX-OS バージョンでは TAM バージョン 3 が使用されていました。

上位リリースから下位リリースへのダウングレード

このセクションには、上位リリースから下位リリースへのダウングレードの例が含まれていません。



注意 後で使用できるように、構成ファイルを必ず保管してください。

ダウングレードするには、次の手順を実行します。

手順

ステップ 1 `write erase` コマンドを入力します。

ステップ 2 `write erase boot` コマンドを入力します。

ステップ 3 `copy Release 6.0(2)U6(3a)-config startup-config` コマンドを入力します。

```
switch# copy downgrade startup-config
```

ステップ 4 `install all kickstart img.kick` システム `img.sys no-save bios-force` コマンドを入力します。

ステップ 5 ブート変数が存在するかどうかを確認するには、`show boot` コマンドを入力します。ブート変数が存在しない場合は、次の一連のコマンドを入力して、ブート変数を手動で更新します。

```
switch# configure t
switch (config)# boot kickstart <img.kick>
switch (config)# boot system <img.sys>
switch (config)# write run start
```

(注) スイッチでロックアップが発生した場合は、`tftp` ブートを実行します。

インストールのトラブルシューティング

アップグレードが失敗する一般的な原因としては、次のようなものがあります。

- `bootflash:` に、更新されたイメージを格納できるだけの十分なスペースがない。
- アップグレードの実行中にハードウェアが取り付けられた、または取り外された。
- アップグレードの実行中に停電があった。
- リモート サーバの場所のパス全体が正確に指定されていない。



第 3 章

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 \(39 ページ\)](#)
- [モジュラ パッケージの使用 \(41 ページ\)](#)
- [Cisco NX-OS ソフトウェア パッケージのリスト \(42 ページ\)](#)
- [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 \(44 ページ\)](#)
- [ISSU のサポート \(44 ページ\)](#)
- [RPM に関する情報 \(44 ページ\)](#)
- [YUM コマンドに関する情報 \(61 ページ\)](#)
- [インストール操作ユーザ ロールの作成 \(80 ページ\)](#)
- [Compacting Cisco NX-OS ソフトウェア イメージの圧縮 \(80 ページ\)](#)

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。

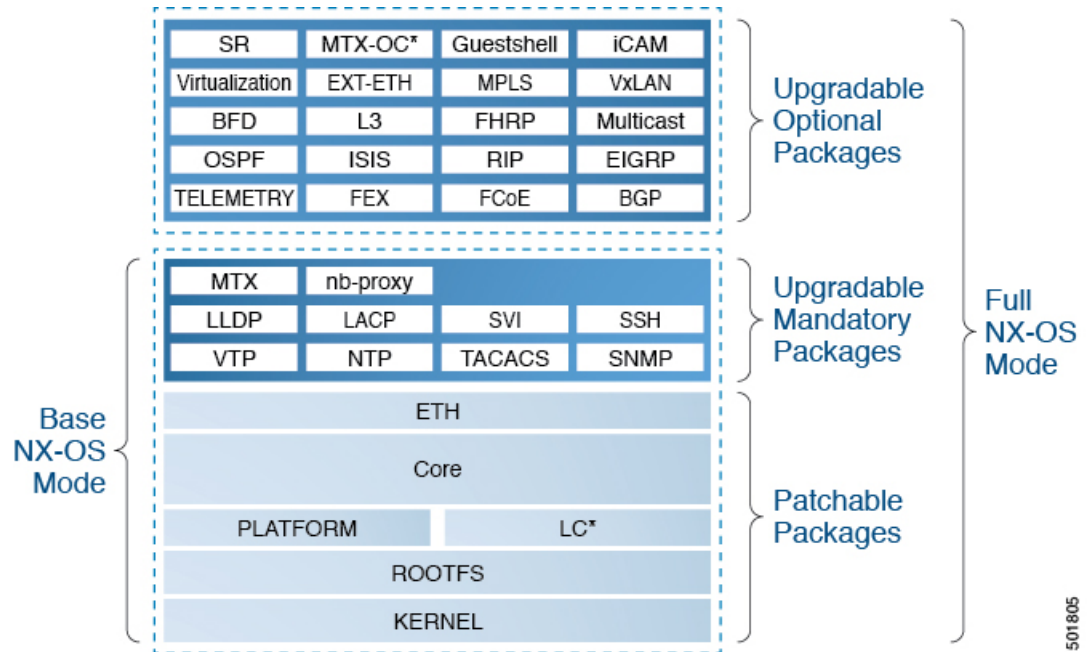
モジュール式の Cisco NX-OS ソフトウェアを使用する利点は次のとおりです。

- 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化が可能です。

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 9: 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"> • feature openflow • feature evb • feature imp • feature netflow • feature sla_sender • feature sla_responder • feature sla twamp-server • feature sflow
FCoE	<ul style="list-style-type: none"> • feature-set fcoe • feature-set fcoe-npv
FEX	feature-set fex
FHRP	<ul style="list-style-type: none"> • feature hsrp • feature vrrpv3
iCAM	feature icam
ISIS	feature isis
MPLS	<ul style="list-style-type: none"> • feature mpls segment-routing • feature mpls evpn
マルチキャスト	<ul style="list-style-type: none"> • feature pim • feature pim6 • feature msdp • feature ngmvpn

パッケージ名	関連機能
OSPF	<ul style="list-style-type: none"> • feature ospf • feature ospfv3
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"> • feature nv overlay • feature fabric forwarding

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 10: 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 install add 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、install add 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、install remove 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に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 11: 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 12:機能 RPM 操作の install CLI リファレンス

CLI	説明
install reset	<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>install reset コマンドも write erase 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>
install reset nxos base	<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（write erase）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>
install reset nxos full	<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（write erase）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>
install add <>	<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（patch/feature/third-party）を更新します。</p>

CLI	説明
install activate <rpm name>	リポジトリに存在する RPM をインストールします。
install commit <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。
install deactivate <rpm name>	RPM をアンインストールします。
install remove <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。
sh install active	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)。
sh install inactive	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。
sh install packages	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
cgrouplite.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>show install packages</p> <p>例 :</p> <pre>switch# show install packages</pre> <p>Boot Image: NXOS Image: bootflash:/nxos.9.2.1.bin</p> <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-udhcp.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+gitaa8187dcd3b7ad67d8e5e3a15115d3eef43a7ed1-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>show install packages grep <lib32_n9000></pre> <p>例 :</p> <pre>switch# show install packages grep lib32_n9000 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> show install packages grep <x86_64> 例 : switch# show install packages grep x86_64 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>install add <rpm>activate</pre> <p>例 :</p> <pre>switch# install add bootflash:chef.rpm activate 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 <rpm></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 <rpm></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	install add <rpm>activate upgrade 例 : <pre>switch(config)# install add bootflash:bp-2.0.2.0-9.2.1.lib32_n9000.rpm activate upgrade</pre> <p>Adding the patch (/bgp-2.0.2.0-9.2.1.lib32_n9000.rpm) [#####] 100% Install operation 870 completed successfully at Tue May 8 11:22:30 2018</p> <p>Activating the patch (/bgp-2.0.2.0-9.2.1.lib32_n9000.rpm) [#####] 100% Install operation 871 completed 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 <rpm>activate downgrade</pre> <p>例 :</p> <pre>switch(config)# install add bootflash:bgp-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> install remove <rpm> 例 : switch(config)# show install inactive grep vxlan vxlan-2.0.0.0-9.2.1.lib32_n9000 switch(config)# install remove vxlan 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 13: 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 ...

```

```

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 ...
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 Version	Arch Size	Repository
vxlan 2.0.1.0-9.2.1	lib32_n9000 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
  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
  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:
  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 Size
Installing:			
bgp	lib32_n9000	groups-repo	2.0.0.0-9.2.1 2.4 M
eigrp	lib32_n9000	groups-repo	2.0.0.0-9.2.1 428 k
isis	lib32_n9000	groups-repo	2.0.0.0-9.2.1 1.2 M
rip	lib32_n9000	groups-repo	2.0.0.0-9.2.1 214 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 Size
Removing:			
bgp	lib32_n9000	@groups-repo	2.0.0.0-9.2.1 11 M
eigrp	lib32_n9000	@groups-repo	2.0.0.0-9.2.1 2.0 M
isis	lib32_n9000	@groups-repo	2.0.0.0-9.2.1 5.7 M
ospf	lib32_n9000	@groups-repo	2.0.0.0-9.2.1 15 M
rip	lib32_n9000	@groups-repo	2.0.0.0-9.2.1 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 14: 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"> • <code>opid</code> : ある操作 ID に固有なログ • <code>last</code> : 最新の操作のログを示します。 • <code>reverse</code> : 逆の順序でログを表示します。 • <code>detail</code> : 詳細ログを表示します。 • <code>from</code> : 特定の操作 ID 以降のログを示します。
<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"> • <code>all</code> : ログを完全にクリアします。 • <code>:</code> : この操作 ID 以降のログをクリアします。
<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> (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。