



Cisco ASR 9000 シリーズ ルータ (IOS XR リリース 7.5.x) システムセットアップおよびソフトウェア インストール ガイド

初版 : 2021 年 11 月 30 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.

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)

© 2021 Cisco Systems, Inc. All rights reserved.



目次

第 1 章	新機能および変更機能に関する情報 1
	新規および変更されたシステムセットアップ機能 1

第 2 章	Cisco ASR 9000 システムの機能 3
	Cisco ASR 9000 製品の概要 3
	仮想マシンベースのルーティングとシステム管理 4
	コマンドモード 5

第 3 章	ルータの起動 7
	ルータの起動 7
	USB を使用したルータの起動 9
	iPXE を使用したルータの起動 11
	root ユーザー クレデンシャルの設定 13
	システム管理コンソールへのアクセス 15
	管理ポートの設定 16
	NTP サーバーとのクロック同期の実行 18

第 4 章	予備チェックの実行 19
	ソフトウェア バージョンの確認 19
	アクティブな VM の確認 20
	ハードウェア モジュールのステータスの確認 22
	ファームウェア バージョンの確認 22
	SDR 情報の確認 24
	インターフェイス ステータスの確認 26

第 5 章	ユーザープロファイルの作成および権限の割り当て	27
	ユーザーグループの作成	29
	XR VM でのユーザーグループの設定	29
	システム管理 VM でのユーザーグループの作成	31
	ユーザーの作成	32
	XR VM でのユーザープロファイルの作成	33
	システム管理 VM でのユーザープロファイルの作成	36
	コマンドルールの作成	38
	データルールの作成	41
	ディザスタリカバリのユーザー名とパスワードの変更	43
	PXE ブートを使用したパスワードの回復	45

第 6 章	システムアップグレードの実行および機能パッケージのインストール	47
	システムのアップグレード	47
	サポートされているソフトウェアのアップグレードまたはダウングレードバージョンの表示	49
	機能のアップグレード	54
	最適化されたインストールイメージのサイズ	55
	準備済みパッケージのインストール	57
	パッケージのインストール	60
	パッケージのアンインストール	64
	プラットフォームでサポートされている機能の表示	66

第 7 章	自動依存関係管理	71
	RPM と SMU の更新	72
	基本ソフトウェアバージョンのアップグレード	73
	RPM のダウングレード	74

第 8 章	ゴールデン ISO を使用したインストールのカスタマイズ	77
	制限事項	78

ゴールデン ISO を使用したインストールのカスタマイズ	79
制限事項	80
ゴールデン ISO ワークフロー	81
ゴールデン ISO の構築	82
スクリプトを使用したゴールデン ISO の構築	83
ゴールデン ISO のインストール	92

第 9 章

クラシック ZTP を使用したルータの展開	99
構成ファイルの作成	101
ユーザースクリプトの作成	101
ZTP シェルユーティリティ	102
ZTP ヘルパー Python ライブラリ	104
データ ポートでの認証	109
DHCP サーバーのセットアップ	110
ZTP 初期化ファイルのカスタマイズ	113
ルータのフレッシュブート時のゼロタッチプロビジョニング	115
DHCP を使用したフレッシュブート	115
手動による ZTP の呼び出し	116



第 1 章

新機能および変更機能に関する情報

次の表に、Cisco ASR 9000 シリーズルータのシステムセットアップおよびソフトウェアインストールガイド [英語] の新機能および変更機能に関する情報の要約を示します。

- [新規および変更されたシステムセットアップ機能 \(1 ページ\)](#)

新規および変更されたシステムセットアップ機能

機能	説明	変更が行われたリリース
プラットフォームでサポートされている機能の表示	この機能には、ルータのリリースでサポートされている機能とサポートされていない機能のリストが表示されます。この機能を使用すると、サポート情報用の注釈付き機能を使用して、ネットワーク設定をより適切に計画できます。 この機能では、 show features コマンドが導入されています。	リリース 7.5.2
サポートされているソフトウェアのアップグレードまたはダウングレードバージョンの表示 (49 ページ)	この機能を使用して、ソフトウェアバージョンを別の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るかどうかを判断できます。実際のアップグレードやダウングレードプロセスを実行する前に、アップグレードやダウングレードが失敗する原因となるハードウェアまたはソフトウェアの制限を確認することもできます。 この機能は、正常なソフトウェアのアップグレードやダウングレードを計画するのに役立ちます。 この機能では、 show install upgrade-matrix コマンドが導入されています。	リリース 7.5.1

機能	説明	変更が行われたリリース
ゴールドデン ISO (GISO) ファイルの完全性の確認	この機能には、md5sum 値を使用して GISO 内のファイルの完全性を検証するオプションがあります。	リリース 7.5.1
強化されたゴールドデン ISO ビルドツール	この拡張機能により、gisobuild.py ツールを使用して、カスタマイズされたインストール要件に合わせて Cisco IOS XR ソフトウェアコマンド、YAML ベースのテンプレートファイル、または Docker 機能を使用して GISO イメージを柔軟に構築できます。GISO を作成する場合は、基本イメージとオプションの RPM を使用してルータを自動的にプロビジョニングすることに加えて、ゼロタッチプロビジョニング (ZTP) 初期化ファイル、スクリプト初期化ファイル、Cisco IOS XR 構成ファイル、および SMU の指定もできます。	リリース 7.5.1
インストール操作に関するテレメトリデータのストリーミング	<p>この機能を使用すると、インストール操作中に検出された変更について、MDT データ (パターン駆動型テレメトリ (CDT) データとイベント駆動型テレメトリ (EDT) データの両方) をストリーミングできます。</p> <p>この機能を使用すると、次のセンサーパスからのデータをストリーミングできます。</p> <pre>Cisco-IOS-XR-install-oper:install/request</pre> <pre>Cisco-IOS-XR-install-oper:install/packages/committed/summary</pre> <pre>Cisco-IOS-XR-install-oper:install/packages/active/summary</pre> <pre>Cisco-IOS-XR-install-oper:install/version</pre>	リリース 7.5.1



第 2 章

Cisco ASR 9000 システムの機能

この章で説明する内容は次のとおりです。

- [Cisco ASR 9000 製品の概要 \(3 ページ\)](#)
- [仮想マシンベースのルーティングとシステム管理 \(4 ページ\)](#)
- [コマンドモード \(5 ページ\)](#)

Cisco ASR 9000 製品の概要

Cisco ASR 9000 シリーズ ルータは、サービス プロバイダー アプリケーション向けに最適化された次世代のエッジアクセスルータです。Cisco ASR 9000 シリーズルータは、次のさまざまな役割を果たすように設計されています。

- レイヤ 2 およびレイヤ 3 イーサネット アグリゲーション
- サブスクリバ対応ブロードバンドアグリゲーション

Cisco ASR 9000 シリーズルータは、冗長性、可用性、パッケージング、電源、およびサービス プロバイダーの従来の要件に関するキャリアクラスの要件を満たしています。

Cisco ASR 9000 シリーズは、次のルータで構成されています。

- Cisco ASR 9001 ルータ (32 ビット)
- Cisco ASR 9001-S ルータ
- Cisco ASR 9006 ルータ
- Cisco ASR 9010 ルータ
- Cisco ASR 9901 ルータ
- Cisco ASR 9904 ルータ
- Cisco ASR 9906 ルータ
- Cisco ASR 9910 ルータ
- Cisco ASR 9912 ルータ

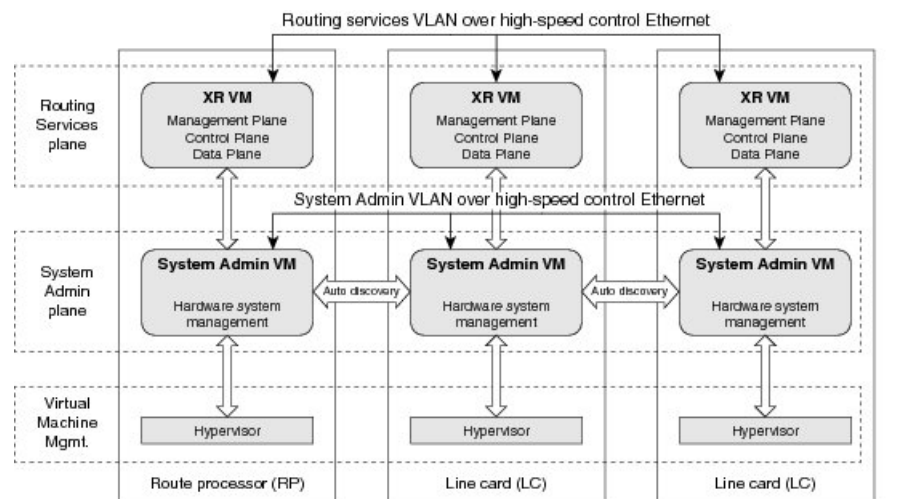
- Cisco ASR 9922 ルータ

仮想マシンベースのルーティングとシステム管理

64 ビット IOS XR を実行している Cisco ASR 9000 シリーズ ルータでは、ルーティング機能とシステム管理機能は、Linux ホスト オペレーティング システム上の個別の仮想マシン (VM) で実行されます。VM は、共通のハードウェア上で個々の物理コンピューティング環境をシミュレートします。プロセッサ、メモリ、ハードディスクなどの使用可能なハードウェアリソースは、ハイパーバイザによって仮想化され、個々の仮想マシンに割り当てられます。

64 ビット IOS XR を実行している Cisco ASR 9000 シリーズ ルータの VM トポロジを次の図に示します。

図 1: Cisco ASR 9000 シリーズ ルータの仮想化された IOS XR



Cisco ASR 9000 シリーズ ルータでの仮想化された IOS XR の実装

- ハイパーバイザにより、個々の VM 環境が作成および管理されます。
- すべてのルートプロセッサ (RP) に 2 つの VM があります。1 つはシステム管理用 (システム管理 VM) で、もう 1 つはルーティング機能 (XR VM) の管理用です。
- 各ノードの 2 つの VM は、それぞれのプレーンで動作します。各プレーンの VM は、高速の制御イーサネット接続を介し、専用の VLAN を使用して相互に接続されます。
- システム管理 VM は、自動検出によって互いの存在を検出できるため、完全なシステム認識を維持できます。

XR VM にアクセスするには、RP の XR VM コンソールポートに接続します。システム管理 VM にアクセスするには、XR VM CLI で **admin** コマンドを実行します。



(注) 32 ビット IOS XR OS では、管理インターフェイスは XR VM から使用できます。64 ビット IOS XR OS では、RP/RSP の管理ポートは次のように使用できます。

- MGT LAN 0 は XR VM で使用できます。
- MGT LAN 1 は管理 VM で使用できます。

ルータ上の仮想化された IOS XR の利点

- 起動時間の短縮：システム管理機能が専用の VM 上にあるため、起動時間が大幅に短縮されます。
- 独立したアップグレード：システム管理 VM と XR VM にソフトウェアパッケージを個別にインストールできるため、システムのダウンタイムが最小限に抑えられます。
- 自己起動 VM：システム管理 VM と XR VM の両方が、ユーザーの介入なしでルータの起動時に自動的に起動されます。すぐに使用できるデフォルトのセットアップがあります。
- システムの冗長性：相互接続性を保ちながら、VM 間がある程度分離されているため、特定の VM で問題が発生しても、他の VM の機能には影響を及ぼしません。

コマンドモード

次の表に、コマンドモードを示します。

コマンドモード	説明
XR VM 実行モード	XR VM でコマンドを実行してルータの動作状態を表示します。 例： RP/0/RP0/CPU0:router#
XR VM グローバル設定	XR VM でセキュリティ、ルーティング、および他の XR 機能の設定を実行します。 例： RP/0/RP0/CPU0:router#configure RP/0/RP0/CPU0:router(config)#

コマンドモード	説明
システム管理 VM 実行モード	<p>システム管理 VM でコマンドを実行して、ルータハードウェアの動作状態を表示およびモニターします。シャーシまたは個別のハードウェア モジュールは、このモードでリロードすることができます。</p> <p>例 :</p> <pre>RP/0/RP0/CPU0:router#admin sysadmin-vm:0_RP0#</pre>
システム管理 VM コンフィギュレーションモード	<p>システム管理 VM でコンフィギュレーション コマンドを実行して、シャーシ全体のハードウェアモジュールを管理および操作します。</p> <p>例 :</p> <pre>RP/0/RP0/CPU0:router#admin sysadmin-vm:0_RP0#config sysadmin-vm:0_RP0 (config)#</pre>



第 3 章

ルータの起動

ハードウェアの設置後、ルータを起動します。XR コンソールポートに接続し、ルータの電源をオンにします。ルータは、プリインストールされたオペレーティングシステム (OS) イメージを使用してブートプロセスを実行します。ルータ内に使用できるイメージがない場合は、PXE ブートまたは外部のブート可能な USB ドライブを使用してルータを起動できます。

起動が完了したら、root ユーザー名とパスワードを作成します。その組み合わせを使って XR コンソールにログインするとルータプロンプトが表示されます。XR コンソールで作成された最初のユーザーは、システム管理コンソールに同期されます。XR コンソールからシステム管理コンソールにアクセスして、システム管理設定を行います。

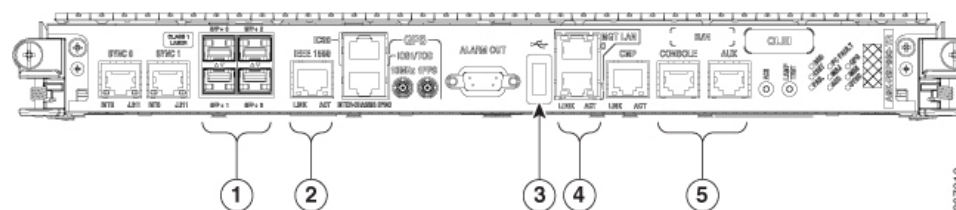
ハードウェア設置の詳細については、『[Cisco ASR 9000 Series Aggregation Services Router Hardware Installation Guide](#)』を参照してください。

大規模な環境で手動による介入なしでルータをリモートでプロビジョニングするには、ゼロタッチプロビジョニング (ZTP) を使用することを推奨します。[クラシック ZTP を使用したルータの展開 \(99 ページ\)](#) を参照してください。

- [ルータの起動 \(7 ページ\)](#)
- [USB を使用したルータの起動 \(9 ページ\)](#)
- [iPXE を使用したルータの起動 \(11 ページ\)](#)
- [root ユーザー クレデンシャルの設定 \(13 ページ\)](#)
- [システム管理コンソールへのアクセス \(15 ページ\)](#)
- [管理ポートの設定 \(16 ページ\)](#)
- [NTP サーバーとのクロック同期の実行 \(18 ページ\)](#)

ルータの起動

新しいルータに接続するには、ルートプロセッサ (RP) のコンソールポートを使用します。コンソールポートはデフォルトでXR コンソールに接続されます。必要に応じて、設定した管理ポートを通じて後続の接続を確立できます。



1	SFP/SFP+ ポート
2	サービス LAN ポート
3	外部 USB ポート
4	管理 LAN ポート
5	コンソールおよび補助 (AUX) ポート

ステップ 1 RP のコンソールポートに端末を接続します。

ステップ 2 ワークステーションで端末エミュレーションプログラムを起動します。

- モジュール型シャーシ RP の場合、コンソール設定はボーレートが 9600 bps、パリティなし、ストップビットが 1、データビットが 8 です。
- 固定シャーシの場合、コンソール設定はボーレートが 115200 bps、パリティなし、ストップビットが 1、データビットが 8 です。

ボーレートはデフォルトで設定されていて、変更することはできません。

RSP4、RP2 カード搭載のシャーシの場合、コンソール設定はボーレートが 9600 bps、パリティなし、ストップビットが 1、データビットが 8 です。ユーザーは、このボーレートを変更できます。次世代の RP3、RSP5 カードの場合、コンソール設定はボーレートが 115200 bps、パリティなし、ストップビットが 1、データビットが 8 です。

ステップ 3 ルータの電源を投入します。

電源コードを電源入力モジュール (PEM) に接続してルータを起動します。端末エミュレーションプログラムのコンソール画面に、ブートプロセスの詳細が表示されます。

ステップ 4 Enter を押します。

root-system ユーザー名の入力を求めるプロンプトが表示されたらブートプロセスは完了です。プロンプトが表示されない場合は、ルータの初期ブート手順が完了するまでしばらく待ってから Enter を押しください。

重要 ブートプロセスが失敗する場合、ルータにプリインストールされているイメージが破損している可能性があります。この場合は、外部のブート可能な USB ドライブを使用してルータを起動できます。

- (注) ルータが新しいバージョンで起動するサーバーに送信元の場所からイメージをコピー後、そのイメージの md5sum を確認することを推奨します。コピーすることで、md5sum の不一致が確認された場合に、破損したファイルを削除し、セットアップを開始するためにイメージファイルの作業コピーを使用できます。

次のタスク

root ユーザー名およびパスワードを指定します。

USB を使用したルータの起動

ブート可能な USB ドライブを使用して、システムアップグレードの目的でルータを再イメージ化したり、起動に失敗した場合にパスワードを復元したり、ルータを起動したりします。ルータの USB はディスク 2 としてマウントされます。

始める前に

次の前提条件を満たしていることを確認します。

- ストレージ容量が 8GB (最小) ~ 32GB (最大) の USB ドライブにアクセスできる。USB 2.0 および USB 3.0 がサポートされています。
- 圧縮ブートファイルを [Software Download Center](#) からローカルマシンにコピーしている。圧縮ブートファイルのファイル名の形式は、`asr9k-x64-usb_boot-<release_number>.zip` です。

ステップ 1 ブート可能な USB ドライブを作成します。

- (注) 圧縮ファイルの内容 (「EFI」および「boot」ディレクトリ) は、USB ドライブのルートに直接展開する必要があります。解凍アプリケーションによって展開ファイルが新しいフォルダに配置された場合は、「EFI」および「boot」ディレクトリを USB ドライブのルートに移動してください。
- a) USB ドライブをローカルマシンに接続し、Windows オペレーティングシステムまたは Apple MAC ディスクユーティリティを使用して FAT32 または MS-DOS ファイルシステムでフォーマットします。
 - b) `asr9k-x64-usb_boot-<release_number>.zip` 圧縮ブートファイルを USB ドライブにコピーします。
 - c) コピー処理が正常に行われたことを確認します。確認するには、コピー元とコピー先でファイルサイズを比較します。さらに、MD5 チェックサム値を確認します。
 - d) 圧縮ブート ファイルを USB ドライブ内で解凍して内容を展開します。これにより、USB ドライブがブート可能なドライブに変換されます。
 - e) ローカル マシンから USB ドライブを取り出します。

ステップ 2 アクティブ RP に USB を挿入し、ルータの電源をリロードまたはリセットします。

(注) この手順はアクティブ RP でのみ使用してください。スタンバイ RP はシャーシから取り外すか、ブートメニューで停止する必要があります。USB のイメージでアクティブ RP がインストールされたら、スタンバイ RP をブートします。

ステップ3 アクティブ XR コンソールで、Ctrl+C キーを押して BIOS メニューを表示します。メニューから [IOS-XR 64 bit Local boot using front panel USB media] を選択します。

```

Got EMT Mode as Disk Boot
Set OS type None, Received OS type=0
Got Boot Mode as Disk Boot

Booting IOS-XR 64 bit Boot previously installed image - Press Ctrl-c to stop
.
Please select the operating system and the boot device:
  1) Boot to ROMMON
  2) IOS-XR 64 bit Boot previously installed image
  3) IOS-XR 64 bit Mgmt Network boot using DHCP server
  4) IOS-XR 64 bit Mgmt Network boot using local settings (iPXE)
  (Press 'p' for more option)
Selection [1/2/3/4]: p
Please select the operating system and the boot device:
  1) Boot to ROMMON
  2) IOS-XR 64 bit Boot previously installed image
  3) IOS-XR 64 bit Mgmt Network boot using DHCP server
  4) IOS-XR 64 bit Mgmt Network boot using local settings (iPXE)
  5) IOS-XR 64 bit Internal network boot from RSP/RP
  6) IOS-XR 64 bit Local boot using embedded USB media
  7) IOS-XR 64 bit Local boot using front panel USB media
  8) Change baud rate and continue booting
Selection [1/2/3/4/5/6/7/8]: 7
Selected IOS-XR 64 bit Local boot using front panel USB media, Continue ? Y/N: y

Set CBC OS type IOS-XR 64 bit, EMT USB Boot to CBC
Sending boot success notification

Selected boot option - EFI USB Device 1 (SanDisk Cruzer)
Verifying image signature...
Image signature verified successfully
Image Verification Passed

```

522185

アクティブおよびスタンバイ RP がブートメニューで停止しない場合は、以前使用されていたブートオプションが使用されます。ブートメニューで 30 分間何も操作しない場合、システムは自動的にリセットされます。

ステップ4 スタンバイ RP が存在し、ステップ2で停止した場合、アクティブ RP の起動開始後にスタンバイ RP を起動します。ブートオプションから、[IOS-XR 64 bit Internal network boot from RSP/RP] を選択します。

例：

```

Please select the operating system and the boot device:
  1) IOS-XR (32 bit Classic XR)
  2) IOS-XR 64 bit Boot previously installed image
  3) IOS-XR 64 bit Mgmt Network boot using DHCP server
  4) IOS-XR 64 bit Mgmt Network boot using local settings (iPXE)
  5) IOS-XR 64 bit Internal network boot from RSP/RP
  6) IOS-XR 64 bit Local boot using embedded USB media
  7) IOS-XR 64 bit Local boot using front panel USB media
Selection [1/2/3/4/5/6/7]:

```


オプション 5 を選択し、起動を続行します。ルータが起動したら、ルートユーザー名とパスワードを指定します。

iPXE を使用したルータの起動

iPXE は、管理インターフェイスのネットワーク カードに含まれ、ルータのシステム ファームウェア (UEFI) レベルで動作するプリブート実行環境です。iPXE は、システムを再イメージするために使用され、ブートに失敗した場合や有効なブート可能なパーティションがない場合にルータを起動します。iPXE は ISO イメージをダウンロードして、イメージのインストールを進行させ、新しいインストール内でブートストラップを行います。

iPXE はブート ロードアとして機能し、システムを起動するイメージをプラットフォーム ID (PID)、シリアル番号、または管理 MAC アドレスに基づいて柔軟に選択できるようにします。iPXE は DHCP サーバーのコンフィギュレーションファイルで定義する必要があります。



- (注) PID とシリアル番号は、コマンド `(admin) hw-module location all bootmedia network reload all` を使用して iPXE を呼び出した場合にのみサポートされます。iPXE を BIOS から手動で選択した場合、PID およびシリアル番号はサポートされません。



- (注) **Cisco ASR 9901** : iPXE のデフォルトでは、以前に試行されたブート方法が次のリロード時に使用されます。[Network] オプションが以前に使用されていた場合、iPXE レジスタは 1 (IPXE_PREF=1) に設定されます。次のリロード時に DHCP 経由で Cisco ASR 9901 ルータを起動するには、IPXE_PREF 設定を 0 (IPXE_PREF=0) に設定する必要があります。

システム管理コンソールから、`run chvrf 0 ssh rp0_admin /opt/cisco/calvados/bin/nvram_dump -s IPXE_PREF=0` コマンドを 2 回入力します。このコマンドを初めて入力すると、ホストが既知のホストリストに追加されます。

```
sysadmin-vm:0_RP0# run chvrf 0 ssh rp0_admin /opt/cisco/calvados/bin/nvram_dump -s
IPXE_PREF=0
Sat May 2 10:39:52.740 UTC+00:00
Warning: Permanently added 'rp0_admin' (ECDSA) to the list of known hosts.
sysadmin-vm:0_RP0# run chvrf 0 ssh rp0_admin /opt/cisco/calvados/bin/nvram_dump -s
IPXE_PREF=0
Sat May 2 10:39:54.995 UTC+00:00
sysadmin-vm:0_RP0# hw-module location all bootmedia network
```

iPXE ブートは、次の場合に実行できます。

- 移行スクリプトを使用した 32 ビットから 64 ビットへの移行
- パスワードの復旧
- 64 ビット イメージを使用したブート障害

始める前に

iPXE ブート後に設定をロードバックできるように、TFP または FTP パスに設定のバックアップを作成します。

ステップ1 システム管理コンソールへのログイン

例：

```
sysadmin-vm:0_RSP0# hw-module location all reload
Tue Mar 6 08:12:47.605 UTC
Reload hardware module ? [no,yes] yes
result Card graceful reload request on all acknowledged.
sysadmin-vm:0_RSP0#
```

ステップ2 ルータをブートできない場合は、次の情報が表示されたときに Ctrl + C キーを押してブートプロセスを停止します。

(注) この手順はアクティブ RP でのみ使用してください。スタンバイ RP はシャーシから取り外すか、ブートメニューで停止する必要があります。iPXE ブートのイメージでアクティブ RP がインストールされたら、スタンバイ RP をブートします。

例：

```
System Bootstrap, Version 10.57 [ASR9K x86 ROMMON],
Copyright (c) 1994-2018 by Cisco Systems, Inc.
Compiled on Mon 01/09/2017 17:15:01.98
BOARD_TYPE           : 0x100317
Rommon               : 10.57 (Primary)
Board Revision       : 4
PCH EEPROM           : 3.4
IPU FPGA (PL)        : 0.40.0 (Backup)
IPU INIT (HW_FPD)    : 0.30.0
IPU FSBL (BOOT.BIN)  : 1.19.0
IPU LINUX (IMAGE.FPD) : 1.21.0
OPTIMUS FPGA         : 0.12.0
OMEGA FPGA           : 0.13.0
ALPHA FPGA           : 0.14.0
CHA FPGA             : 0.5.1
CBC0                 : Part 1=34.38, Part 2=34.38, Act Part=2
Product Number       : A9K-RSP880-SE
Chassis              : ASR-9904-AC
Chassis Serial Number : FOX1936GBDD
Slot Number          : 1
Pxe Mac Address LAN 0 : 70:e4:22:06:13:40
Pxe Mac Address LAN 1 : 70:e4:22:06:13:41
=====
Got EMT Mode as 3
Got Boot Mode as 0
Booting IOS-XR (32 bit Classic XR) - Press Ctrl-c to stop
```

ステップ3 iPXE ブートの場合は、オプション 4 を選択します。

例：

```
Please select the operating system and the boot device:
 1) IOS-XR (32 bit Classic XR)
 2) IOS-XR 64 bit Boot previously installed image
 3) IOS-XR 64 bit Mgmt Network boot using DHCP server
 4) IOS-XR 64 bit Mgmt Network boot using local settings (iPXE)
```

```
5) IOS-XR 64 bit Internal network boot from RSP/RP
6) IOS-XR 64 bit Local boot using embedded USB media
7) IOS-XR 64 bit Local boot using front panel USB media
Selection [1/2/3/4/5/6/7]:
```

ステップ4 FTP または TFTP を使用してブートする前に、iPXE ROMMON の詳細を手動で更新します。

例：

```
iPXE>set cisco/cisco-server-url:string tftp://<path>/asr9k-mini-x64.iso
iPXE>set cisco/cisco-ipv4-address:string 1.3.24.202
iPXE>set cisco/cisco-netmask-address:str 255.255.0.0
iPXE>set cisco/cisco-gateway-address:str 1.3.0.1
```

ステップ5 接続されている管理ポート (0/1) を開きます。

例：

```
iPXE>ifclose net0
iPXE>ifclose net1
iPXE>ifopen net1
```

ここで、net0 と net1 は管理 port0 と port1 をそれぞれ表します。

ステップ6 FTP または TFTP の場所から必要なイメージをブートします。

例：

```
iPXE>
iPXE> ifopen net0:
iPXE> boot tftp://<path>/asr9k-mini-x64-<release-number>.iso
tftp://<path>/asr9k-mini-x64-<release-number>.iso... 0%
Booting iso-image@0x83c525000 (1135456256), bzImage@0x83c55f000 (4526671)

.....BIOS CODE SIGN ENTRY ...
```

ステップ7 アクティブ RP が稼働したら、スタンバイ RP をブートします。ブートオプションから、[IOS-XR 64 bit Internal network boot from RSP/RP] を選択します。

例：

```
Please select the operating system and the boot device:
1) IOS-XR (32 bit Classic XR)
2) IOS-XR 64 bit Boot previously installed image
3) IOS-XR 64 bit Mgmt Network boot using DHCP server
4) IOS-XR 64 bit Mgmt Network boot using local settings (iPXE)
5) IOS-XR 64 bit Internal network boot from RSP/RP
6) IOS-XR 64 bit Local boot using embedded USB media
7) IOS-XR 64 bit Local boot using front panel USB media
Selection [1/2/3/4/5/6/7]:
```

root ユーザー クレデンシャルの設定

ルータの初回起動時に、ルートログイン情報（ユーザー名とパスワード）の設定を求めるプロンプトが表示されます。ルートログイン情報は、XR コンソール（root-lr）、システム管理 VM（root-system）のルートユーザー用、およびディザスタリカバリの目的で設定されています。

ステップ 1 Enter root-system username: *username*

root ユーザーのユーザー名を入力します。文字数制限は 1023 文字です。この例では、root ユーザーの名前は「root」です。

重要 指定したユーザー名は、XR コンソールの「root-lr」グループにマッピングされます。また、システム管理コンソールの「root-system」ユーザーとしてもマッピングされます。

ルータの初回起動時、またはルータのオペレーティングシステムをデフォルト状態にリセット後は、そのルータにユーザー設定はありません。この場合、ルータによって「root-system ユーザー名」を指定するように要求されます。ただしすでにルータが設定されている場合は、ステップ 4 で説明したように「ユーザー名」の入力を求めるプロンプトが表示されます。

ステップ 2 Enter secret: *password*

root ユーザーのパスワードを入力します。パスワードの文字数は 6 ～ 253 文字です。セキュリティ上の理由から、入力したパスワードは CLI に表示されません。

root-system ユーザーにはスーパーユーザー権限があるため、root-system ユーザー名とパスワードは保護する必要があります。root-system ユーザーはルータ設定全体へのアクセスに使用されます。

ステップ 3 Enter secret again: *password*

root-system ユーザーのパスワードをもう一度入力します。セキュリティ上の理由から、入力したパスワードは CLI に表示されません。

ステップ 4 Username: *username*

root-system ユーザー名を入力して、XR VM コンソールにログインします。

ステップ 5 Password: *password*

root-system ユーザーのパスワードを入力します。正しいパスワードを入力するとルータのプロンプトが表示されます。これで XR VM コンソールにログインできました。

ステップ 6 (任意) show run username

ユーザーの詳細を表示します。

```
username root
group root-lr
group cisco-support
secret 5 $1$NBg7$fHs1inKPZVvzqxMv775UE/
!
```

次のタスク

- XR コンソールからルーティング機能を設定します。
- システム管理プロンプトでシステム管理設定を行います。システム管理プロンプトは、システム管理コンソールへのアクセス時に表示されます。システム管理プロンプトを表示す

る方法については、[システム管理コンソールへのアクセス（15 ページ）](#) を参照してください。

システム管理コンソールへのアクセス

すべてのシステム管理とハードウェア管理の設定を実行するには、XR コンソールからシステム管理コンソールにログインする必要があります。

ステップ 1 ルートユーザーとして XR コンソールにログインします。

ステップ 2 （任意）XR モードからシステム管理モードにアクセスする場合は、コンソール ポートでログイン バナーを無効にします。

- a) **configure**
- b) **service sysadmin-login-banner disable**

例：

```
RP/0/RSP0/cpu 0: router(config)#service sysadmin-login-banner disable
```

システム管理モードでコンソール ポートのログイン バナーを無効にします。

- c) **commit**
- d) **end**

ステップ 3 **admin**

例：

ログインバナーは、デフォルトで有効に設定されています。次の例では、ログインバナーを有効にした状態のコマンド出力を示しています。

```
RP/0/RSP0/cpu 0: router#admin
```

```
Mon May 22 06:57:29.350 UTC
```

```
root connected from 127.0.0.1 using console on host
```

```
sysadmin-vm:0_RP0# exit
```

```
Mon May 22 06:57:32.360 UTC
```

次の例では、ログインバナーを無効にした状態のコマンド出力を示しています。

```
RP/0/RP0/CPU0:router#admin
```

```
Thu Mar 01:07:14.509 UTC
```

```
sysadmin-vm:0_RP0# exit
```

ステップ 4 （任意） **exit**

システム管理モードから XR モードに戻ります。

管理ポートの設定

管理ポートをシステム管理およびリモート通信に使用するには、管理イーサネットインターフェイスの IP アドレスとサブネット マスクを設定する必要があります。他のネットワーク上のデバイス（リモート管理ステーションや TFTP サーバーなど）と通信する場合は、ルータのデフォルト（スタティック）ルートを設定する必要があります。

始める前に

- ネットワーク管理者またはシステムの設計担当者にお問い合わせ、管理インターフェイスの IP アドレスおよびサブネット マスクを入手します。
- RP の物理ポートイーサネット 0 とイーサネット 1 は管理ポートです。ポートが管理ネットワークに接続されていることを確認します。

手順の概要

1. **configure**
2. **interface MgmtEth rack/slot/port**
3. **ipv4 address ipv4-address subnet-mask**
4. **ipv4 address ipv4 virtual address subnet-mask**
5. **no shutdown**
6. **exit**
7. **router static address-family ipv4 unicast 0.0.0.0/0 default-gateway**
8. **commit** または **end** コマンドを使用します。

手順の詳細

ステップ 1 **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

ステップ 2 **interface MgmtEth rack/slot/port**

例：

```
RP/0/RSP0/cpu 0: router(config)#interface mgmtEth 0/RP0/CPU0/0
```

プライマリ RP の管理インターフェイスのインターフェイス コンフィギュレーション モードを開始します。

ステップ 3 **ipv4 address ipv4-address subnet-mask**

例：

```
RP/0/RSP0/cpu 0: router(config-if)#ipv4 address 10.1.1.1/8
```

IP アドレスとサブネット マスクをインターフェイスに割り当てます。

ステップ 4 `ipv4 address ipv4 virtual address subnet-mask`

例 :

```
RP/0/RSP0/cpu 0: router(config-if)#ipv4 address 1.70.31.160 255.255.0.0
```

仮想 IP アドレスとサブネット マスクをインターフェイスに割り当てます。

ステップ 5 `no shutdown`

例 :

```
RP/0/RSP0/cpu 0: router(config-if)#no shutdown
```

インターフェイスを「アップ」状態にします。

ステップ 6 `exit`

例 :

```
RP/0/RSP0/cpu 0: router(config-if)#exit
```

管理インターフェイス コンフィギュレーション モードを終了します。

ステップ 7 `router static address-family ipv4 unicast 0.0.0.0/0 default-gateway`

例 :

```
RP/0/RSP0/cpu 0: router(config)#router static address-family ipv4 unicast 0.0.0.0/0 12.25.0.1
```

デフォルト ゲートウェイの IP アドレスを指定して、スタティック ルートを設定します。このルートは他のネットワーク上のデバイスと通信する際に使用します。

ステップ 8 `commit` または `end` コマンドを使用します。

commit : 設定の変更を保存し、コンフィギュレーションセッションに留まります。

end : 次のいずれかのアクションの実行をユーザーに要求します。

- **Yes** : 設定の変更を保存し、コンフィギュレーションセッションを終了します。
- **No** : 設定の変更をコミットせずに、コンフィギュレーションセッションを終了します。
- **Cancel** : 設定の変更をコミットせずに、コンフィギュレーションセッションに留まります。

次のタスク

管理ポート経由でイーサネット ネットワークに接続します。端末エミュレーション プログラムで、管理インターフェイス ポートへの SSH または Telnet 接続をその IP アドレスを使って確立します。ルータに対して許可される Telnet セッションの数を設定するには、Telnet セッションを確立する前に、XR コンフィギュレーション モードで `telnet ipv4|ipv6 server max-servers` コマンドを使用します。

NTP サーバーとのクロック同期の実行

XR コンソールとシステム管理コンソールにはそれぞれのシステム クロックがあります。これらのクロックが実際の時間とずれないように、NTP サーバーのクロックと同期する必要があります。このタスクでは、XR コンソール用に NTP サーバーを設定します。XR コンソールのクロックを同期すると、システム管理コンソールのクロックは自動的に XR コンソールのクロックと同期されます。

始める前に

管理ポートを設定して接続します。

ステップ 1 `configure`

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

ステップ 2 `ntp server server_address`

例：

```
RP/0/RSP0/cpu 0: router(config)#ntp server 64.90.182.55
```

指定したサーバーと同期するように XR コンソールのクロックが設定されます。



第 4 章

予備チェックの実行

コンソールに正常にログインしたら、予備チェックを実行してデフォルト設定を確認する必要があります。チェックの実行時に設定の問題が検出された場合は、さらに設定を行う前に修正を行ってください。予備チェックの内容は次のとおりです。

- [ソフトウェアバージョンの確認 \(19 ページ\)](#)
- [アクティブな VM の確認 \(20 ページ\)](#)
- [ハードウェア モジュールのステータスの確認 \(22 ページ\)](#)
- [ファームウェアバージョンの確認 \(22 ページ\)](#)
- [SDR 情報の確認 \(24 ページ\)](#)
- [インターフェイス ステータスの確認 \(26 ページ\)](#)

ソフトウェアバージョンの確認

ルータには、プリインストールされた Cisco IOS XR ソフトウェアが付属しています。ソフトウェアの最新バージョンがインストールされていることを確認します。新しいバージョンを使用できる場合は、システムアップグレードを実行してください。これにより新しいバージョンのソフトウェアがインストールされ、ルータに最新の機能セットが提供されます。

ルータで実行されている Cisco IOS XR ソフトウェアのバージョンを確認するには、次のタスクを実行します。

手順の概要

1. show version

手順の詳細

show version

例：

```
RP/0/RSP0/cpu 0: router# show version
```

ルータにインストールされている各種ソフトウェア コンポーネントのバージョンを表示します。結果には、Cisco IOS XR ソフトウェアとその各種コンポーネントのバージョンが含まれます。

例

次のタスク

結果を確認して、システム アップグレードまたは追加のパッケージ インストールが必要かどうかを特定します。必要な場合は、「[システムアップグレードの実行および機能パッケージのインストール](#)」の章のタスクを参照してください。

アクティブな VM の確認

ルータでは XR VM とシステム管理 VM の両方が動作している必要があります。両方の VM のインスタンスは、すべてのルートプロセッサ (RP) で実行されている必要があります。VM がアクティブになっているか確認するには、次のタスクを実行します。

手順の概要

1. **show redundancy summary**
2. **admin**
3. **show vm**

手順の詳細

ステップ 1 show redundancy summary

例：

```
RP/0/RP0:hostname#show redundancy summary
Mon Mar 9 16:32:19.276 IST
Active Node Standby Node
-----
0/RP0 0/RP1 (Node Ready, NSR: Not Configured)
0/LC0 0/LC1 (Node Ready, NSR: Not Configured)
RP/0/RP0:hostname#
```

VM の準備状況を表示します。

ステップ 2 admin

例：

```
RP/0/RSP0/cpu 0: router# admin
```

管理 EXEC モードを開始します。

ステップ3 show vm

例：

```
sysadmin-vm:0_RP0#show vm
```

さまざまなノードで実行されている VM のステータスを表示します。

```
sysadmin-vm:0_RP0# sh vm
Mon Mar 9 07:52:06.173 UTC
----- VMs found at location 0/RP0 -----
Id : sysadmin
Status : running
IP Addr: 192.0.44.1
HB Interval : NA
Last HB Sent: NA
Last HB Rec : NA
-----
Id : default-sdr
Status : running
IP Addr: 192.0.44.4
HB Interval : 0 s 500000000 ns
Last HB Sent: 663743
Last HB Rec : 663743
-----
Id : default-sdr
Status : running
IP Addr: 192.0.44.6
HB Interval : 10 s 0 ns
Last HB Sent: 33183
Last HB Rec : 33183
-----
----- VMs found at location 0/RP1 -----
Id : sysadmin
Status : running
IP Addr: 192.0.88.1
HB Interval : NA
Last HB Sent: NA
Last HB Rec : NA
-----
Id : default-sdr
Status : running
IP Addr: 192.0.88.4
HB Interval : 0 s 500000000 ns
Last HB Sent: 663749
Last HB Rec : 663749
-----
Id : default-sdr
Status : running
IP Addr: 192.0.88.6
HB Interval : 10 s 0 ns
Last HB Sent: 33183
Last HB Rec : 33183
-----
sysadmin-vm:0_RP0#
```

上記の結果について説明します。

- Id : VM の名前。「sysadmin」はシステム管理 VM を表します。「default-sdr」は XR VM を表します。
- Status : VM のステータス。
- IP Addr : VM の内部 IP アドレス。

VM がノードで実行されていない場合、**show vm** コマンドの出力には、そのノードに関する出力は表示されません。

次のタスク

XR VM がノードで実行されていない場合は、ノードをリロードしてください。リロードするには、モードで **hw-module location node-id reload** コマンドを使用します。また、モードで **show sdr** コマンドを使用して、SDR がノードで実行されていることを確認します。

ハードウェア モジュールのステータスの確認

ハードウェア モジュールには RP、ファントレイなどがあります。ルータには複数のハードウェア モジュールが取り付けられています。すべてのハードウェア モジュールが正しく取り付けられて動作していることを確認するには、次のタスクを実行します。

始める前に

必要なハードウェア モジュールがすべてルータに取り付けられていることを確認します。

ファームウェア バージョンの確認

ルータのさまざまなハードウェア コンポーネントのファームウェアは、インストールされている Cisco IOS XR イメージと互換性がある必要があります。互換性がないと、ルータの誤動作を引き起こす可能性があります。ファームウェア バージョンを確認するには、次のタスクを実行します。

手順の概要

1. show hw-module fpd

手順の詳細

show hw-module fpd

例：

ルータで検出されたハードウェア モジュールのリストを表示します。

(注) このコマンドは、XR VM とシステム管理 VM の両方のモードで実行できます。

上記の出力で重要なフィールドは次のとおりです。

- FPD Device : FPD、CFP などのハードウェア コンポーネントの名前。
- ATR : ハードウェア コンポーネントの属性。次のような属性があります。

- B : バックアップ イメージ
- S : セキュア イメージ
- P : 保護されたイメージ

- Status : ファームウェアのアップグレード ステータス。それぞれの状態については次のとおりです。
 - CURRENT : ファームウェア バージョンは最新バージョンです。
 - READY : FPD のファームウェアはアップグレード可能な状態です。
 - NOT READY : FPD のファームウェアはアップグレード可能な状態ではありません。
 - NEEDUPGD : インストール済みのイメージで新しいファームウェア バージョンを利用できます。アップグレードすることが推奨されます。
 - RLOAD REQ : アップグレードが完了しており、ISO イメージのリロードが必要です。
 - UPGD DONE : ファームウェア アップグレードが正常に行われました。
 - UPGD FAIL : ファームウェア アップグレードが失敗しました。
 - BACKIMG : ファームウェアが破損しています。ファームウェアを再インストールしてください。
 - UPGD SKIP : インストール済みファームウェアのバージョンが、イメージで利用可能なバージョンよりも上位であるため、アップグレードがスキップされました。

- Running : FPD で現在実行中のファームウェアのバージョン。

次のタスク

- EXEC モードで **upgrade hw-module location all fpd** コマンドを使用して、必要なファームウェアをアップグレードします。FPD アップグレードを有効にするには、ルータの電源を再投入する必要があります。
- **upgrade hw-module fpd all location {all | node-id}** コマンドを使用して、特定のノードの FPGA をすべてアップグレードすることを推奨します。**upgrade hw-module fpd <individual-fpd> location {all | node-id}** を使用してノードの FPGA をアップグレードしないでください。カードの起動時にエラーが発生する可能性があります。
- 必要に応じて、自動 FPD アップグレード機能を有効にします。有効にするには、XR コンフィギュレーション [(config)#] モードで **fpd auto-upgrade enable** コマンドを使用します。有効にすると、ルータにインストールされているイメージに新しい FPD バイナリが存在する場合、システムのアップグレード処理中に FPD が自動的にアップグレードされます。

SDR 情報の確認

セキュアドメインルータ（SDR）は、単一の物理システムを論理的に独立した複数のルータに分割します。SDR は論理ルータ（LR）とも呼ばれます。ルータでは 1 つの SDR のみがサポートされます。この SDR をデフォルト SDR と呼びます。すべてのルータには、ルーティングシステムにインストールされている RP をすべて所有するデフォルト SDR が付属しています。この SDR のインスタンスは、ラインカードおよびルートプロセッサで実行されます。SDR インスタンスの詳細を確認するには、次のタスクを実行します。

手順の概要

1. **admin**
2. **show sdr**

手順の詳細

ステップ 1 admin

例：

```
RP/0/RSP0/cpu 0: router# admin
```

管理 EXEC モードを開始します。

ステップ 2 show sdr

例：

```
sysadmin-vm:0_RP0# show sdr
```

各ノードの SDR 情報が表示されます。

```
sysadmin-vm:0_RP0# show sdr
```

```
sdr default-sdr
location 0/0/VM1
sdr-id                2
IP Address of VM     192.0.4.3
MAC address of VM    A4:6C:2A:2B:AA:A6
VM State              RUNNING
start-time            2015-12-03T15:38:38.74514+00:00
Last Reload Reason   "SMU:Reboot triggered by install"
Reboot Count         2
location 0/1/VM1
sdr-id                2
IP Address of VM     192.0.8.3
MAC address of VM    B0:AA:77:E7:5E:DA
VM State              RUNNING
start-time            2015-12-03T15:38:39.730036+00:00
Last Reload Reason   "SMU:Reboot triggered by install"
Reboot Count         2
location 0/2/VM1
sdr-id                2
IP Address of VM     192.0.12.3
MAC address of VM    B0:AA:77:E7:67:34
```

```
VM State          RUNNING
start-time       2015-12-03T15:38:38.886947+00:00
Last Reload Reason "SMU:Reboot triggered by install"
Reboot Count     2
location 0/3/VM1
sdr-id           2
IP Address of VM 192.0.16.3
MAC address of VM B0:AA:77:E7:58:86
VM State         RUNNING
start-time       2015-12-03T15:38:40.391205+00:00
Last Reload Reason "SMU:Reboot triggered by install"
Reboot Count     2
location 0/4/VM1
sdr-id           2
IP Address of VM 192.0.20.3
MAC address of VM B0:AA:77:E7:46:C2
VM State         RUNNING
start-time       2015-12-03T15:38:39.84469+00:00
Last Reload Reason "SMU:Reboot triggered by install"
Reboot Count     2
location 0/5/VM1
sdr-id           2
IP Address of VM 192.0.24.3
MAC address of VM B0:AA:77:E7:84:40
VM State         RUNNING
start-time       2015-12-04T03:48:24.017443+00:00
Last Reload Reason "VM_REQUESTED_UNGRACEFUL_RELOAD:Headless SDR"
Reboot Count     3
location 0/6/VM1
sdr-id           2
IP Address of VM 192.0.28.3
MAC address of VM B0:AA:77:E7:55:FE
VM State         RUNNING
start-time       2015-12-03T15:38:38.74753+00:00
Last Reload Reason "SMU:Reboot triggered by install"
Reboot Count     2
location 0/7/VM1
sdr-id           2
IP Address of VM 192.0.32.3
MAC address of VM B0:AA:77:E7:60:C6
VM State         RUNNING
start-time       2015-12-03T15:38:38.691481+00:00
Last Reload Reason "SMU:Reboot triggered by install"
Reboot Count     2
location 0/RP0/VM1
sdr-id           2
IP Address of VM 192.0.108.4
MAC address of VM 10:05:CA:D7:FE:6F
VM State         RUNNING
start-time       2015-12-04T07:03:04.549294+00:00
Last Reload Reason CARD_SHUTDOWN
Reboot Count     1
location 0/RP1/VM1
sdr-id           2
IP Address of VM 192.0.112.4
MAC address of VM 10:05:CA:D8:3F:43
VM State         RUNNING
start-time       2015-12-04T09:21:42.083046+00:00
Last Reload Reason CARD_SHUTDOWN
Reboot Count     1
```

機能 SDR では、VM の状態は「RUNNING」です。SDR がノードで動作していない場合、結果の該当箇所
に出力が表示されません。

次のタスク

SDR がノードで動作していない場合は、ノードのリロードを試してください。このためには、
モードで **hw-module location node-id reload** コマンドを使用します。

インターフェイスステータスの確認

ルータが起動すると、使用可能なすべてのインターフェイスがシステムによって検出されま
す。インターフェイスが検出されない場合、ユニットの異常を示している可能性があります。
検出されたインターフェイスの数を確認するには、次のタスクを実行します。

手順の概要

1. show ipv4 interface summary

手順の詳細

show ipv4 interface summary

例：

```
RP/0/RSP0/cpu 0: router#show ipv4 interface summary
```

ルータの初回起動時には、すべてのインターフェイスが「未割り当て」の状態です。結果に表示されるイ
ンターフェイスの総数が、ルータに存在するインターフェイスの実際の数と一致することを確認してくだ
さい。

上記の結果について説明します。

- **Assigned** : IP アドレスがインターフェイスに割り当てられています。
- **Unnumbered** : ルータの他のインターフェイスにすでに設定された IP アドレスを借用しているインター
フェイスです。
- **Unassigned** : IP アドレスはインターフェイスに割り当てられていません。

また、モードで **show interfaces brief** および **show interfaces summary** コマンドを使用し、インターフェイ
スステータスを確認することもできます。



第 5 章

ユーザープロファイルの作成および権限の割り当て

ルータ上の XR およびシステム管理設定へのアクセス権を管理するには、権限を割り当てたユーザープロファイルを作成します。権限はコマンドルールとデータルールを使用して指定します。

ユーザー、グループ、コマンドルール、およびデータルールを作成するには、認証、許可、およびアカウントिंग (AAA) コマンドを使用します。aaa コマンドはディザスタリカバリパスワードを変更する際にも使用します。



(注) 外部 AAA サーバーおよびサービスは、システム管理 VM からは設定できず、XR VM からのみ設定できます。

制御されていないアクセスをユーザーが行わないよう制限するために、AAA 認証を設定します。AAA 認証が設定されていない場合、ユーザーに割り当てられたグループに関連付けられたコマンドおよびデータルールはバイパスされます。IOS-XR ユーザーは、ネットワーク設定プロトコル (NETCONF)、Google 定義のリモートプロシージャコール (gRPC) または任意の YANG ベースのエージェントを介して、IOS-XR 設定への完全な読み取り/書き込みアクセス権を持つことができます。制御されていないアクセスを許可しないようにするには、いずれかの設定を行う前に AAA 認証を有効にします。

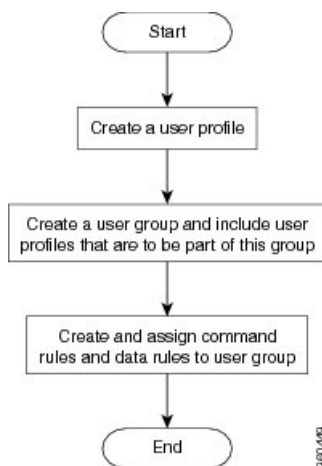


- (注) XR 上のいずれかのユーザーが削除されている場合、ローカルデータベースは、システム管理 VM に最初のユーザーが存在するかどうかを確認します。
- 最初のユーザーが存在する場合、同期は実行されません。
 - 最初のユーザーが存在しない場合は、XR の最初のユーザー（作成順序に基づく）がシステム管理 VM に同期されます。
 - ユーザーが XR に追加され、システム管理モードにユーザーが存在しない場合、そのユーザーは `sysadmin-vm` に同期されます。同期後、XR VM のユーザーに対する変更は、システム管理 VM では同期されません。
 - システム管理 VM に追加されたユーザーが XR VM と同期しない。
 - システム管理 VM で作成された最初のユーザーまたはディザスタリカバリユーザーのみがホスト VM と同期します。
 - システム管理 VM の最初のユーザーまたはディザスタリカバリユーザーのログイン情報の変更は、ホスト VM と同期されます。
 - システム管理 VM で削除された最初のユーザーまたはディザスタリカバリユーザーは、ホスト VM と同期されません。ホスト VM でユーザーが保持されます。

ユーザーの認証にはユーザー名とパスワードが使用されます。認証されたユーザーは、ユーザーグループに対して作成および適用されているコマンドルールとデータルールに基づいて、コマンドを実行しデータ要素にアクセスする権利が与えられます。ユーザーグループに属するすべてのユーザーには、そのユーザーグループのコマンドルールおよびデータルールで定義されているシステムへのアクセス権があります。

ユーザープロファイルを作成するためのワークフローを次のフローチャートに示します。

図 2: ユーザープロファイル作成のワークフロー





- (注) ルータの初回起動時に作成された XR VM の root-lr ユーザーは、システム管理 VM の root-system ユーザーにマッピングされます。root-system ユーザーにはシステム管理 VM のスーパーユーザー権限があるため、アクセスは制限されません。

既存の AAA 設定を表示するには、コンフィギュレーションモードで **show run aaa** コマンドを使用します。

この章で説明する内容は次のとおりです。

- [ユーザーグループの作成 \(29 ページ\)](#)
- [ユーザーの作成 \(32 ページ\)](#)
- [コマンドルールの作成 \(38 ページ\)](#)
- [データルールの作成 \(41 ページ\)](#)
- [ディザスタリカバリのユーザー名とパスワードの変更 \(43 ページ\)](#)
- [PXE ブートを使用したパスワードの回復 \(45 ページ\)](#)

ユーザーグループの作成

新しいユーザーグループを作成してコマンドルールとデータルールを関連付けます。コマンドルールおよびデータルールは、ユーザーグループに属するすべてのユーザーに適用されます。

ユーザーグループ、タスクグループ、RADIUS および TACACS 設定の作成の詳細については、*System Security Configuration Guide for Cisco ASR 9000 Series Routers* の「AAA サービスの設定」の章を参照してください。コマンド、構文、および構文の説明の詳細については、*System Security Command Reference for Cisco ASR 9000 Series Routers* の「認証、許可、およびアカウントिंगコマンド」の章を参照してください。

XR VM でのユーザーグループの設定

ユーザーグループは、タスクグループなど一連のユーザーに対するコマンドパラメータによって設定されます。**usergroup** コマンドを入力すると、ユーザーグループコンフィギュレーションサブモードが開始されます。**usergroup** コマンドの **no** 形式を使用すると、特定のユーザーグループを削除できます。システムで参照されているユーザーグループを削除すると、警告が表示されます。

始める前に



- (注) WRITE:AAA タスク ID が関連付けられているユーザーだけ、ユーザーグループを設定できません。ユーザーグループは、事前定義されたグループのプロパティ (owner-sdr など) を継承できません。

ステップ 1 configure

例 :

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

ステップ 2 usergroup *usergroup-name*

例 :

```
RP/0/RSP0/cpu 0: router(config)# usergroup beta
```

特定のユーザー グループの名前を作成し、ユーザー グループ コンフィギュレーション サブモードを開始します。

- **usergroup** コマンドの **no** 形式を指定すると、特定のユーザー グループをシステムから削除できます。

ステップ 3 description *string*

例 :

```
RP/0/RSP0/cpu 0: router(config-ug)#  
description this is a sample user group description
```

(任意) ステップ 2 で指定したユーザーグループの説明を作成します。

ステップ 4 inherit usergroup *usergroup-name*

例 :

```
RP/0/RSP0/cpu 0: router(config-ug)#  
inherit usergroup sales
```

- ユーザー グループの権限を明示的に定義します。

ステップ 5 taskgroup *taskgroup-name*

例 :

```
RP/0/RSP0/cpu 0: router(config-ug)# taskgroup beta
```

ステップ 2 で指定したユーザー グループをこのステップで指定したタスク グループに関連付けます。

- ユーザー グループは、入力したタスク グループに対してすでに定義されている設定属性 (タスク ID リストと権限) を取ります。

ステップ 6 ステップ 2 で指定したユーザー グループを関連付ける各タスク グループに対して手順を繰り返します。

ステップ 7 commit または **end** コマンドを使用します。

commit : 設定の変更を保存し、コンフィギュレーションセッションに留まります。

end : 次のいずれかのアクションの実行をユーザーに要求します。

- **Yes** : 設定の変更を保存し、コンフィギュレーションセッションを終了します。

- **No** : 設定の変更をコミットせずに、コンフィギュレーションセッションを終了します。
- **Cancel** : 設定の変更をコミットせずに、コンフィギュレーションセッションに留まります。

システム管理 VM でのユーザーグループの作成

システム管理 VM のユーザーグループを作成します。

ルータでは、最大 32 のユーザーグループがサポートされます。

始める前に

ユーザープロファイルを作成します。「ユーザーの作成」の項を参照してください。

ステップ 1 admin

例 :

```
RP/0/RSP0/cpu 0: router# admin
```

管理 EXEC モードを開始します。

ステップ 2 config

例 :

```
sysadmin-vm:0_RP0#config
```

モードを開始します。

ステップ 3 aaa authentication groups group group_name

例 :

```
sysadmin-vm:0_RP0(config)#aaa authentication groups group gr1
```

新しいユーザーグループ（まだ存在していない場合）を作成して、グループコンフィギュレーションモードを開始します。この例では、ユーザーグループ「gr1」が作成されます。

(注) デフォルトで、root ユーザーの作成時にユーザーグループ「root-system」がシステムによって作成されます。root ユーザーはこのユーザーグループのメンバです。このグループに追加されたユーザーは root ユーザー権限を取得します。

ステップ 4 users user_name

例 :

```
sysadmin-vm:0_RP0(config-group-gr1)#users us1
```

ユーザーグループに含めるユーザーの名前を指定します。

複数のユーザー名を二重引用符で囲んで指定することができますたとえば、**users "user1 user2 ..."**となります。

ステップ 5 `gid group_id_value`

例 :

```
sysadmin-vm:0_RP0(config-group-gr1)#gid 50
```

数値を指定します。32 ビットの整数を入力できます。

ステップ 6 `commit` または `end` コマンドを使用します。**commit** : 設定の変更を保存し、コンフィギュレーションセッションに留まります。**end** : 次のいずれかのアクションの実行をユーザーに要求します。

- **Yes** : 設定の変更を保存し、コンフィギュレーションセッションを終了します。
- **No** : 設定の変更をコミットせずに、コンフィギュレーションセッションを終了します。
- **Cancel** : 設定の変更をコミットせずに、コンフィギュレーションセッションに留まります。

次のタスク

- コマンドルールを作成します。
- データルールを作成します。

ユーザーの作成

新しいユーザーを作成し、特定の権限を持つユーザーグループにそのユーザーを含めることができます。ルータでは、最大で 1024 個のユーザープロファイルがサポートされます。



- (注) システム管理 VM で作成したユーザーは、XR VM で作成したユーザーとは異なるため、システム管理 VM ユーザーのユーザー名とパスワードを使用して XR VM にアクセスすることはできません。逆も同様です。

XR VM およびシステム管理 VM ユーザー プロファイルの同期

ユーザープロファイルの初期同期 : ユーザープロファイルを XR VM で初めて作成した場合、そのユーザーがシステム管理 VM に存在しない場合のみ、ユーザー名とパスワードがシステム管理 VM に同期されます。この初期同期により、2つの VM 間でのユーザー情報の一貫性が確保されます。

後続の変更の制限 : ただし、システム管理 VM では、XR VM 内で行われた後続のパスワード変更やユーザーの削除は同期されないことに注意することが重要です。その結果、XR VM とシステム管理 VM のパスワードが異なり、XR VM 内での削除を反映するためにユーザープロファイルがリアルタイムで更新されない場合があります。

ユーザーの削除処理：さらに、XR VM 内でユーザーが削除されても、システム管理 VM 内の対応するユーザープロファイルは影響を受けません。つまり、XR VM でユーザーを削除しても、システム管理 VM のユーザープロファイルは自動的に削除されません。

ユーザーグループ、タスクグループ、RADIUS および TACACS 設定の作成の詳細については、*System Security Configuration Guide for Cisco ASR 9000 Series Routers* の「AAA サービスの設定」の章を参照してください。コマンド、構文、および構文の説明の詳細については、*System Security Command Reference for Cisco ASR 9000 Series Routers* の「認証、許可、およびアカウントिंगコマンド」の章を参照してください。

XR VM でのユーザープロファイルの作成

表 1: 機能の履歴 (表)

機能名	リリース情報	機能説明
拡張ログインバナーの標準規格	リリース 7.3.1	US DoD に準拠するために、ログインバナーの表示を有効にするオプションが導入されました。ログインバナーは、成功したログイン試行回数と失敗したログイン試行回数、タイムスタンプ、ログイン方法などの情報を提供します。 login-history コマンドが導入されました。

各ユーザーは、管理ドメイン内で一意のユーザー名によって識別されます。各ユーザーは、少なくとも 1 つのユーザーグループのメンバーである必要があります。ユーザーグループを削除すると、そのグループに関連付けられたユーザーが孤立する場合があります。AAA サーバーでは孤立したユーザーも認証されますが、ほとんどのコマンドは許可されません。

AAA の詳細については、*System Security Configuration Guide for Cisco ASR 9000 Series Routers* の「AAA サービスの設定」の章を参照してください。関連コマンド、構文、および構文の説明の詳細については、*System Security Command Reference for Cisco ASR 9000 Series Routers* の「認証、許可、およびアカウントिंगコマンド」の章を参照してください。

ステップ 1 configure

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

ステップ 2 username user-name

例：

```
RP/0/RSP0/cpu 0: router(config)# username user1
```

新しいユーザーの名前を作成（または現在のユーザーを識別）して、ユーザー名コンフィギュレーションサブモードを開始します。

- **user-name** 引数には 1 つの単語だけ使用できます。スペースと引用符は使用できません。

ステップ 3 次のいずれかを実行します。

- **password {0 | 7} password**
- **secret {0 | 5 | 8 | 9 | 10} secret**

例：

```
Router(config-un)# password 0 pwd1
```

または

```
Router(config-un)# secret 0 secl
```

ステップ 2 で指定したユーザーのパスワードを指定します。

- **secret** コマンドを使用して、ステップ 2 で指定したユーザー名用の安全なログインパスワードを作成します。
- **password** コマンドの後に **0** を入力すると、暗号化されていない（クリアテキスト）パスワードが続くことが指定されます。**password** コマンドの後に **7** を入力すると、暗号化されたパスワードが続くことが指定されます。
- **secret** コマンドでは、次の値を入力できます。
 - **0**：セキュアな暗号化されていない（クリアテキスト）パスワードが続くことを指定します。
 - **5**：MD5 ハッシュアルゴリズムを使用するセキュアな暗号化パスワードが続くことを指定します。
 - **8**：SHA256 ハッシュアルゴリズムを使用するタイプ 8 シークレットが続くことを指定します。
 - **9**：scrypt ハッシュアルゴリズムを使用するタイプ 9 シークレットが続くことを指定します。

(注) タイプ 8 およびタイプ 9 のシークレットは、Cisco IOS XR ソフトウェアリリース 7.0.1 以降の IOS XR 64 ビットオペレーティングシステムでサポートされています。リリース 7.0.1 より前は、IOS XR 32 ビットオペレーティングシステムでのみサポートされていました。
 - **10**：SHA512 ハッシュアルゴリズムを使用するタイプ 10 シークレットを指定します。

- (注)
- タイプ 10 シークレットは、Cisco IOS XR 64 ビットプラットフォームでのみサポートされています。
 - まだ **MD5** または **SHA256** 暗号化アルゴリズムを使用している下位バージョンにダウングレードする場合、設定の損失、認証の失敗など、後方互換性の問題が発生することが予想されます。タイプ 10 シークレットがある場合、システムをバージョン 7.0.1 以降からバージョン 6.5.3 以降にダウングレードする場合は、**シークレット** をタイプ 5 に変換します。システムをバージョン 7.0.1 以降から 6.5.3 未満のバージョンにダウングレードする場合は、**install activate** を実行する前に、XR-vm および **sysadmin-vm** からすべてのユーザーの設定を解除します。Cisco IOS XR 32 ビットソフトウェアを実行している Cisco ASR 9000 シリーズルータには、タイプ 10 シークレットが適用されないため、後方互換性の問題は発生しません。
 - 最初のユーザー設定シナリオの場合やユーザーを再設定する場合は、タイプ 5 およびタイプ 10 シークレットのみが XR VM からシステム管理 VM とホスト VM に同期されます。このようなシナリオでは、タイプ 8 およびタイプ 9 シークレットは同期されません。
- タイプ 0 が、**password** コマンドおよび **secret** コマンドのデフォルトです。
 - Cisco IOS XR ソフトウェアリリース 7.0.1 以降では、設定でタイプを選択せずにクリアテキストシークレットが設定されている場合、デフォルトのハッシュタイプは 10 (SHA512) です。

ステップ 4 `group group-name`

例：

```
RP/0/RSP0/cpu 0: router(config-un)# group sysadmin
```

ステップ 2 で指定したユーザーを **usergroup** コマンドで定義したユーザーグループに割り当てます。

- ユーザーは、ユーザーグループのさまざまなタスクグループへの割り当てによって定義された内容に従って、ユーザーグループのすべての属性を受け取ります。
- 各ユーザーは、少なくとも 1 つのユーザーグループに割り当てする必要があります。ユーザーは複数のユーザーグループに属することがあります。

ステップ 5 ステップ 2 で指定したユーザーに関連付けるユーザーグループごとに、ステップ 4 を繰り返します。

ステップ 6 (任意) 米国国防総省 (DoD) 承認済みログインバナーの表示を有効にできます。バナーは、デバイスへのアクセスを許可する前に表示されます。バナーにより、適用される連邦法に準拠したプライバシーとセキュリティの確保も行われます。さらに、システムは、システムブートから、またはユーザープロファイルが作成された直後から、ログインの追跡を行います。

(注) ルータをリロードすると、ログイン通知がリセットされます。

次のコマンドを使用して、ログインバナーを有効または無効にします。

例：

```
Router(config-un)#login-history enable
Router(config-un)#login-history disable
```

show running-config username user1 コマンドを実行して、ログインバナーの状態を確認します。

```
Router(config-un)# show running-config username NAME1
Fri Jan 29 13:55:28.261 UTC
username NAME1
  group UG1
  secret * *****
  password * *****
  login-history enable
```

ステップ 7 **commit** または **end** コマンドを使用します。

commit : 設定の変更を保存し、コンフィギュレーションセッションに留まります。

end : 次のいずれかのアクションの実行をユーザーに要求します。

- **Yes** : 設定の変更を保存し、コンフィギュレーションセッションを終了します。
- **No** : 設定の変更をコミットせずに、コンフィギュレーションセッションを終了します。
- **Cancel** : 設定の変更をコミットせずに、コンフィギュレーションセッションに留まります。

システム管理 VM でのユーザープロファイルの作成

システム管理 VM の新しいユーザーを作成します。ユーザーはユーザー グループに含まれ、特定の権限が割り当てられます。ユーザーには、割り当てられた権限に基づいて、システム管理 VM コンソールのコマンドと設定への制限付きアクセス権が付与されます。

ルータでは、最大で 1024 個のユーザープロファイルがサポートされます。

XR VM の root-lr ユーザーは、EXEC モードで **Admin** コマンドを入力することで、システム管理 VM にアクセスできます。ルータではユーザー名とパスワードの入力を求めるプロンプトは表示されません。XR VM の root-lr ユーザーには、システム管理 VM へのフルアクセス権が付与されます。

ステップ 1 **admin**

例 :

```
RP/0/RSP0/cpu 0: router# admin
```

管理 EXEC モードを開始します。

ステップ 2 **config**

例 :

```
sysadmin-vm:0_RP0#config
```

モードを開始します。

ステップ 3 **aaa authentication users user user_name**

例 :

```
sysadmin-vm:0_RP0(config)#aaa authentication users user us1
```

新しいユーザーを作成し、ユーザー コンフィギュレーション モードを開始します。例では、ユーザー「us1」が作成されます。

ステップ 4 **password** *password*

例 :

```
sysadmin-vm:0_RP0(config-user-us1)#password pwd1
```

システム管理 VM へのログイン時にユーザー認証に使用するパスワードを入力します。

ステップ 5 **uid** *user_id_value*

例 :

```
sysadmin-vm:0_RP0(config-user-us1)#uid 100
```

数値を指定します。32 ビットの整数を入力できます。

ステップ 6 **gid** *group_id_value*

例 :

```
sysadmin-vm:0_RP0(config-user-us1)#gid 50
```

数値を指定します。32 ビットの整数を入力できます。

ステップ 7 **ssh_keydir** *ssh_keydir*

例 :

```
sysadmin-vm:0_RP0(config-user-us1)#ssh_keydir dir1
```

英数字の値を指定します。

ステップ 8 **homedir** *homedir*

例 :

```
sysadmin-vm:0_RP0(config-user-us1)#homedir dir2
```

英数字の値を指定します。

ステップ 9 **commit** または **end** コマンドを使用します。

commit : 設定の変更を保存し、コンフィギュレーションセッションに留まります。

end : 次のいずれかのアクションの実行をユーザーに要求します。

- **Yes** : 設定の変更を保存し、コンフィギュレーションセッションを終了します。
- **No** : 設定の変更をコミットせずに、コンフィギュレーションセッションを終了します。
- **Cancel** : 設定の変更をコミットせずに、コンフィギュレーションセッションに留まります。

コマンドルールの作成

コマンドルールとは、ユーザーグループ内のどのユーザーが特定のコマンドの使用を許可または拒否されるかに基づいたルールです。コマンドルールはユーザーグループに関連付けられ、そのユーザーグループに属するすべてのユーザーに適用されます。

コマンドでの動作を許可するか拒否するかを指定することで、コマンドルールを作成します。次の表に、有効な動作と権限の組み合わせを示します。

動作	承認権限	拒否権限
読み取り (R)	「?」を使用した場合に CLI にコマンドが表示されます。	「?」を使用した場合に CLI にコマンドが表示されません。
実行 (X)	CLI からコマンドを実行できます。	CLI からコマンドを実行できません。
読み取りおよび実行 (RX)	コマンドが CLI に表示され、実行可能です。	コマンドは CLI に表示されず、実行することもできません。

デフォルトでは、すべての権限が **Reject** に設定されています。

各コマンドルールは、関連付けられている番号によって識別されます。ユーザーグループに複数のコマンドルールを適用すると、より小さい番号のコマンドルールが優先されます。たとえば `cmdrule 5` は読み取りアクセスを許可しますが、`cmdrule 10` は読み取りアクセスを拒否するとします。これら両方のコマンドルールを同じユーザーグループに適用すると、`cmdrule 5` が優先されるため、このグループのユーザーは読み取りアクセス権を持ちます。

このタスクの例として、「`show platform`」コマンドの読み取りおよび実行権限を拒否するルールを作成します。

始める前に

ユーザーグループを作成します。[システム管理 VM でのユーザーグループの作成 \(31 ページ\)](#) を参照してください。

手順の概要

1. `admin`
2. `config`
3. `aaa authorization cmdrules cmdrule command_rule_number`
4. `command command_name`
5. `ops {r | x | rx}`
6. `action {accept | accept_log | reject}`
7. `group user_group_name`
8. `context connection_type`
9. `commit` または `end` コマンドを使用します。

手順の詳細

ステップ 1 **admin**

例 :

```
RP/0/RSP0/cpu 0: router# admin
```

管理 EXEC モードを開始します。

ステップ 2 **config**

例 :

```
sysadmin-vm:0_RP0#config
```

モードを開始します。

ステップ 3 **aaa authorization cmdrules cmdrule *command_rule_number***

例 :

```
sysadmin-vm:0_RP0(config)#aaa authorization cmdrules cmdrule 1100
```

コマンドルール番号として数値を指定します。32 ビットの整数を入力できます。

重要 1 ~ 1000 の数字はシスコで予約済みのため使用しないでください。

このコマンドによって、新しいコマンドルール（まだ存在していない場合）が作成され、コマンドルールコンフィギュレーションモードが開始されます。例では、コマンドルール「1100」が作成されます。

(注) デフォルトでは、**root-system** ユーザーの作成時に「**cmdrule 1**」がシステムによって作成されます。このコマンドルールは、すべてのコマンドの「読み取り」および「実行」動作に対する「承認」権限を提供します。したがって「**cmdrule 1**」が変更されない限り、**root** ユーザーに課せられる制限はありません。

ステップ 4 **command *command_name***

例 :

```
sysadmin-vm:0_RP0(config-cmdrule-1100)#command "show platform"
```

権限を制御するコマンドを指定します。

command にアスタリスク「*」を入力した場合、そのコマンドルールがすべてのコマンドに適用されることを意味します。ステップ 5 **ops {r | x | rx}**

例 :

```
sysadmin-vm:0_RP0(config-cmdrule-1100)#ops rx
```

権限を指定する必要がある動作を指定します。

- **r** : 読み取り
- **x** : 実行

- **rx** : 読み取りおよび実行

ステップ 6 **action** {**accept** | **accept_log** | **reject**}

例 :

```
sysadmin-vm:0_RP0(config-cmdrule-1100)#action reject
```

ユーザーがその動作の使用を許可されるか拒否されるかを指定します。

- **accept** : ユーザーはその動作の実行を許可されます。
- **accept_log** : ユーザーはその動作の実行を許可され、アクセスの試行がすべて記録されます。
- **reject** : ユーザーはその動作の実行を制限されます。

ステップ 7 **group** *user_group_name*

例 :

```
sysadmin-vm:0_RP0(config-cmdrule-1100)#group gr1
```

コマンド ルールを適用するユーザーグループを指定します。

ステップ 8 **context** *connection_type*

例 :

```
sysadmin-vm:0_RP0(config-cmdrule-1100)#context *
```

このルールを適用する接続タイプを指定します。接続タイプは *netconf* (ネットワーク設定プロトコル)、*cli* (コマンドライン インターフェイス)、または *xml* (Extensible Markup Language) です。アスタリスク「*」の入力が推奨されます。これは、そのコマンド ルールがすべての接続タイプに適用されることを示します。

ステップ 9 **commit** または **end** コマンドを使用します。

commit : 設定の変更を保存し、コンフィギュレーションセッションに留まります。

end : 次のいずれかのアクションの実行をユーザーに要求します。

- **Yes** : 設定の変更を保存し、コンフィギュレーションセッションを終了します。
- **No** : 設定の変更をコミットせずに、コンフィギュレーションセッションを終了します。
- **Cancel** : 設定の変更をコミットせずに、コンフィギュレーションセッションに留まります。

次のタスク

データ ルールを作成します。 [データ ルールの作成 \(41 ページ\)](#) を参照してください。

データ ルールの作成

データルールとは、ユーザーグループ内のどのユーザーが設定データ要素へのアクセスとその変更を許可または拒否されるかに基づいたルールです。データルールはユーザーグループに関連付けられます。データルールは、ユーザーグループに属するすべてのユーザーに適用されます。

各データルールは、関連付けられている番号によって識別されます。ユーザーグループに複数のデータルールを適用すると、より小さい番号のデータルールが優先されます。

始める前に

ユーザーグループを作成します。[システム管理 VM でのユーザーグループの作成 \(31 ページ\)](#) を参照してください。

手順の概要

1. **admin**
2. **config**
3. **aaa authorization datarules datarule *data_rule_number***
4. **keypath *keypath***
5. **ops *operation***
6. **action {**accept** | **accept_log** | **reject**}**
7. **group *user_group_name***
8. **context *connection type***
9. **namespace *namespace***
10. **commit** または **end** コマンドを使用します。

手順の詳細

ステップ 1 **admin**

例：

```
RP/0/RSP0/cpu 0: router# admin
```

管理 EXEC モードを開始します。

ステップ 2 **config**

例：

```
sysadmin-vm:0_RP0#config
```

モードを開始します。

ステップ 3 **aaa authorization datarules datarule *data_rule_number***

例：

```
sysadmin-vm:0_RP0(config)#aaa authorization datarules datarule 1100
```

データ ルール番号として数値を指定します。32 ビットの整数を入力できます。

重要 1 ~ 1000 の数字はシスコで予約済みのため使用しないでください。

このコマンドによって、新しいデータルール（まだ存在していない場合）が作成され、データルールコンフィギュレーションモードが開始されます。例では、データルール「1100」が作成されます。

(注) デフォルトで、**root-system** ユーザーの作成時に「**datarule 1**」がシステムによって作成されます。このデータルールは、すべての設定データの「読み取り」、「書き込み」、および「実行」動作に対する「承認」権限を提供します。したがって「**datarule 1**」が変更されない限り、**root** ユーザーに課せられる制限はありません。

ステップ 4 **keypath** *keypath*

例 :

```
sysadmin-vm:0_RP0(config-datarule-1100)#keypath /aaa/disaster-recovery
```

データ要素のキーパスを指定します。キーパスはデータ要素の場所を定義する式です。**keypath** にアスタリスク「*」を入力した場合、そのコマンドルールがすべての設定データに適用されることを意味します。

ステップ 5 **ops** *operation*

例 :

```
sysadmin-vm:0_RP0(config-datarule-1100)#ops rw
```

権限を指定する必要がある動作を指定します。各動作は次の文字で識別されます。

- **c** : 作成
- **d** : 削除
- **u** : 更新
- **w** : 書き込み（作成、更新、および削除の組み合わせ）
- **r** : 読み込み
- **x** : 実行

ステップ 6 **action** {**accept** | **accept_log** | **reject**}

例 :

```
sysadmin-vm:0_RP0(config-datarule-1100)#action reject
```

ユーザーがその動作を許可されるか拒否されるかを指定します。

- **accept** : ユーザーはその動作の実行を許可されます。
- **accept_log** : ユーザーはその動作の実行を許可され、アクセスの試行がすべて記録されます。
- **reject** : ユーザーはその動作の実行を制限されます。

ステップ 7 `group user_group_name`

例 :

```
sysadmin-vm:0_RP0(config-datarule-1100)#group gr1
```

データ ルールを適用するユーザー グループを指定します。複数のグループ名を指定することもできます。

ステップ 8 `context connection type`

例 :

```
sysadmin-vm:0_RP0(config-datarule-1100)#context *
```

このルールを適用する接続タイプを指定します。接続タイプは *netconf* (ネットワーク設定プロトコル)、*cli* (コマンドライン インターフェイス)、または *xml* (Extensible Markup Language) です。アスタリスク「*」の入力が推奨されます。これは、そのコマンドがすべての接続タイプに適用されることを示します。

ステップ 9 `namespace namespace`

例 :

```
sysadmin-vm:0_RP0(config-datarule-1100)#namespace *
```

アスタリスク「*」を入力して、データ ルールが名前空間の値すべてに適用されることを示します。

ステップ 10 `commit` または `end` コマンドを使用します。

commit : 設定の変更を保存し、コンフィギュレーションセッションに留まります。

end : 次のいずれかのアクションの実行をユーザーに要求します。

- **Yes** : 設定の変更を保存し、コンフィギュレーションセッションを終了します。
- **No** : 設定の変更をコミットせずに、コンフィギュレーションセッションを終了します。
- **Cancel** : 設定の変更をコミットせずに、コンフィギュレーションセッションに留まります。

ディザスタ リカバリのユーザー名とパスワードの変更

ルータの起動後、最初に `root-system` ユーザー名とパスワードを定義すると、同じユーザー名とパスワードがシステム管理コンソールのディザスタ リカバリ ユーザー名およびパスワードとしてマッピングされます。ただし、これらは変更可能です。

ディザスタ リカバリ ユーザー名およびパスワードは、次の状況で役立ちます。

- システム管理コンソールでの認証のデフォルト ソースである AAA データベースが破損した場合にシステムへアクセスする。
- 何らかの理由でシステム管理コンソールが機能しない場合に、管理ポートを通じてシステムにアクセスする。

- 通常のユーザー名およびパスワードを忘れた場合に、ディザスタリカバリユーザー名とパスワードを使用してシステム管理コンソールにアクセスし、新しいユーザーを作成する。



(注) ルータでは、ディザスタリカバリユーザー名およびパスワードを一度に1つのみ設定できません。

手順の概要

1. **admin**
2. **config**
3. **aaa disaster-recovery username *username* password *password***
4. **commit** または **end** コマンドを使用します。

手順の詳細

ステップ 1 admin

例 :

```
RP/0/RSP0/cpu 0: router# admin
```

管理 EXEC モードを開始します。

ステップ 2 config

例 :

```
sysadmin-vm:0_RP0#config
```

モードを開始します。

ステップ 3 aaa disaster-recovery username *username* password *password*

例 :

```
sysadmin-vm:0_RP0(config)#aaa disaster-recovery username us1 password pwd1
```

ディザスタリカバリユーザー名とパスワードを指定します。既存のユーザーをディザスタリカバリユーザーとして選択する必要があります。この例では、ディザスタリカバリユーザーとして「us1」が選択され、パスワード「pwd1」が割り当てられます。パスワードは、プレーンテキストまたは MD5 ダイジェスト文字列として入力することができます。

ディザスタリカバリユーザー名を使用する場合は、***username*@localhost** の形式で入力してください。

ステップ 4 commit または end コマンドを使用します。

commit : 設定の変更を保存し、コンフィギュレーションセッションに留まります。

end : 次のいずれかのアクションの実行をユーザーに要求します。

- **Yes** : 設定の変更を保存し、コンフィギュレーションセッションを終了します。
- **No** : 設定の変更をコミットせずに、コンフィギュレーションセッションを終了します。
- **Cancel** : 設定の変更をコミットせずに、コンフィギュレーションセッションに留まります。

PXE ブートを使用したパスワードの回復

ログインできない場合、または XR およびシステム管理パスワードを紛失した場合は、次の手順を使用して新しいパスワードを作成します。紛失したパスワードは回復できません。代わりに、新しいユーザー名とパスワードを非グレースフル PXE ブートで作成する必要があります。

ステップ 1 PXE を使用してルータを起動します。

(注) PXE ブートは完全に侵入型なので、ルータの状態、設定、およびイメージがリセットされます。

ルータを PXE ブートするには、[iPXE を使用したルータの起動 \(11 ページ\)](#) を参照してください。

ステップ 2 パスワードのリセット。



第 6 章

システムアップグレードの実行および機能パッケージのインストール

システムアップグレードおよびパッケージインストールプロセスを実行するには、ルータで **install** コマンドを使用します。これらのプロセスでは、iso イメージ (.iso)、機能パッケージをルータ上で追加およびアクティブ化します。ネットワークサーバーからこれらのファイルにアクセスし、ルータ上でアクティブ化します。インストールしたパッケージまたは SMU が原因でルータに問題が発生した場合は、アンインストールすることができます。

この章で説明する内容は次のとおりです。

- システムのアップグレード (47 ページ)
- サポートされているソフトウェアのアップグレードまたはダウングレードバージョンの表示 (49 ページ)
- 機能のアップグレード (54 ページ)
- 最適化されたインストールイメージのサイズ (55 ページ)
- 準備済みパッケージのインストール (57 ページ)
- パッケージのインストール (60 ページ)
- パッケージのアンインストール (64 ページ)
- プラットフォームでサポートされている機能の表示 (66 ページ)

システムのアップグレード

システムのアップグレードとは、ルータに新しいバージョンの Cisco IOS XR オペレーティングシステムをインストールするプロセスです。ルータには Cisco IOS XR イメージがプリインストールされています。ただし、ルータ機能を最新の状態に保つために新しいバージョンをインストールすることができます。システムアップグレード操作は XR VM から実行しますが、システムアップグレードの間に、XR VM とシステム管理 VM の両方で実行しているソフトウェアがアップグレードされます。



(注) ルータ上のインターフェイスに関する設定がなく、**no-shut** 操作を実行して起動した場合、ルータのリロード時にインターフェイスの状態が自動的に **admin-shutdown** に変更されます。



- (注)
- 十分なディスク容量があることを確認します。
 - **fsck** コマンドを実行してファイルシステムのステータスを確認し、IOS XR が正常にアップグレードされていることを確認します。**fsck** コマンドは、システム管理パッケージをインストールする場合はシステム管理 EXEC モードで、XR パッケージをインストールする場合は XR EXEC モードで実行する必要があります。
 - すべての **install** コマンドは、システム管理 EXEC モードと XR EXEC モードの両方で適用できます。システム管理のインストール操作は XR EXEC モードで実行します。

システムアップグレードは、基本パッケージ (Cisco IOS XR ユニキャストルーティング コアバンドル) をインストールして実行します。このバンドルをインストールするには、**install** コマンドを実行します。Cisco IOS XR ユニキャストルーティング コアバンドルのファイル名は、*asr9k-mini-x.iso* です。



注意 ルータのリロード時はインストール操作を実行しないでください。
アップグレード操作中はルータをリロードしないでください。



(注) CSMサーバーは、Web ベースのサーバー側の自動化およびオーケストレーションフレームワークです。サービスプロバイダーは、クリックするだけの簡単な Web インターフェイスを使用して、複数の SMU を同時にスケジュールして展開し、何百台ものルータでソフトウェアのアップグレードをスケジュールどおりに実行できます。詳細については、[Cisco Software Manager](#) を参照してください。



(注) **install commit** コマンドを使用してトランザクションを完了せずに手動または自動のシステムリロードを実行すると、インストールトランザクションが開始される前の状態 (設定の変更を含む) にシステムが復元されます。デバッグ用に保存されるのはログのみです。

このアクションにより、使用可能なすべてのコンフィギュレーション ロールバック ポイントがクリアされます。インストールロールバックイベントが発生するまでは、実行されたコミットにロールバックしたり、コミットを表示したりはできません。インストール ロールバックイベント後に実行された新しいコミットは、コミット ID 1000000001 から開始されます。



- (注) ISSUを使用してシャーシを古いソフトウェアバージョンから IOS XR リリース 7.6.x 以降にアップグレード後、ハードウェアプログラミングを有効にするには、シャーシのリロードを開始します。該当するインターフェイスで最大伝送ユニット (MTU) 値の 9646 を有効にする必要がある場合、シャーシのリロードは必須です。

サポートされているソフトウェアのアップグレードまたはダウングレードバージョンの表示

表 2: 機能の履歴 (表)

機能名	リリース情報	説明
サポートされている IOS XR ソフトウェアのアップグレードまたはダウングレードバージョン	リリース 7.5.1	この機能を使用して、ソフトウェアバージョンを別の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るかどうかを判断できます。実際のアップグレードやダウングレードプロセスを実行する前に、アップグレードやダウングレードが失敗する原因となるハードウェアまたはソフトウェアの制限を確認することもできます。この機能は、正常なソフトウェアのアップグレードやダウングレードを計画するのに役立ちます。 この機能では、 show install upgrade-matrix コマンドが導入されています。

Cisco ルータには、IOS XR ソフトウェアがプリインストールされています。ソフトウェアリリースをアップグレードして新機能とソフトウェア修正を使用するか、ソフトウェアをダウングレードします。追加された新機能または提供されているソフトウェア修正を活用するには、ルータを最新バージョンにアップグレードすることが重要です。

この機能を使用すると、シスコ認定のアップグレードおよびダウングレードパスに適合する Cisco IOS XR ソフトウェアリリースを選択するための次の質問に対する回答を得られます。

- 現在のリリースでサポートされているアップグレードまたはダウングレードリリースは何か。

- リリース X からリリース Y にアップグレードする予定です。使用中のルータではリリース Y へのアップグレードがサポートされていますか。
- ソフトウェアをアップグレードする前にインストールする必要があるブリッジング SMU はありますか。

この機能には、現在のリリースがターゲットリリースへのアップグレードをサポートしているかどうかを判断するメカニズムがあります。このタスクは、**install replace** コマンドを使用した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開始時に実行されます。検証に失敗すると、ソフトウェアのアップグレードはブロックされ、失敗の理由が表示されます。この機能を使用すると、特定のリリースにアップグレードまたはダウングレードできるかどうかをプロアクティブに検査でき、ソフトウェアの計画とアップグレードにかかる時間と労力を節約できます。

この機能は、特定のソフトウェアのアップグレードまたはダウングレードに関連する前提条件や制限事項を理解するのに役立つ次の情報を提供します。

- 必要なブリッジング SMU RPM
- SMU RPM のブロッキング
- サポートされていないハードウェア
- 注意事項または制限事項

install replace コマンドで **force** キーワードを使用すると、自動検証を上書きできます。このオプションを使用すると、アップグレードが失敗したときに警告メッセージが表示されますが、ソフトウェアのアップグレードはブロックされません。**force?** キーワードを使用して、ソフトウェアのアップグレードまたはダウングレードでサポートされるリリースを決定するこのプロセスの無効化以外のシステム機能に対する他の影響を理解します。

次の **show** コマンドを使用するか、運用データからサポート情報を表示できます。

コマンド	説明
show install upgrade-matrix running	実行中のシステムにインストールされているサポートデータに従って、現在のバージョンからのサポートされている全ソフトウェアアップグレードを表示します。
show install upgrade-matrix iso path-to-ISO	実行中のシステムと ISO イメージの両方のサポートデータに従って、現在のバージョンからターゲット ISO のバージョンへのソフトウェアのアップグレードに関する詳細を表示します。
show install upgrade-matrix iso path-to-ISO all	ターゲット ISO イメージのサポートデータに従って、任意のバージョンからのサポートされている全ソフトウェアアップグレードを表示します。

コマンド	説明
show install upgrade-matrix iso path-to-ISO from-running	実行中のシステムとターゲット ISO イメージの両方のサポートマトリックスに従って、現在のバージョンから ISO のバージョンへのソフトウェアのアップグレードに関する詳細を表示します。

実行中のバージョンからのサポートされている全ソフトウェアアップグレードの表示

次に、ルータの現在のバージョン7.5.1からのアップグレードでサポートされている全リリースの例を示します。

```
Router#show install upgrade-matrix running
```

```
Fri Jul 29 10:18:43.413 IST
```

```
This may take a while ...
```

```
The current software [7.5.1] can be upgraded from and downgraded to the following releases:
```

```
=====
```

From	To	Bridge SMUs Required	Caveats
7.0.2	7.5.1	r702.CSCvw57276	None
6.5.3	7.5.1	r653.CSCvw57276	None
7.5.1	7.0.2	None	None
7.5.1	6.5.3	None	- https://www.cisco.com/c/en/us/td/docs/routers/asr9000/software/asr9k-r7-4/system-security/configuration/guide/b-system-security-cg-asr9000-74x/configuring-aaa-services.html#id_128191
7.5.1	7.4.1	None	None
7.5.1	7.1.25	None	None
7.5.1	7.1.3	None	None
7.5.1	7.1.2	None	None
7.5.1	7.3.1	None	None
7.5.1	6.6.3	None	- https://www.cisco.com/c/en/us/td/docs/routers/asr9000/software/asr9k-r7-4/system-security/configuration/guide/b-system-security-cg-asr9000-74x/configuring-aaa-services.html#id_128191
7.5.1	7.3.2	None	None
7.4.1	7.5.1	None	None

```
=====
```

7.1.25	7.5.1	None	None
7.1.3	7.5.1	None	None
7.1.2	7.5.1	None	None
7.3.1	7.5.1	None	None
6.6.3	7.5.1	r663.CSCvw57276	None
7.3.2	7.5.1	None	None

現在のバージョンからターゲットバージョンにソフトウェアをアップグレードするためにサポートされているリリースの表示

次に、ソフトウェアを現在のバージョンからターゲットバージョンにアップグレードするためにサポートされているリリースの例を示します。

```
Router#show install upgrade-matrix iso /harddisk:/asr9k-goldenk9-x64-7.5.2-rev1.iso
```

```
Fri Jul 29 10:19:24.185 IST
This may take a while ...
Upgrade from the current software [7.5.1] to 7.5.2 is supported
=====
From      To        Bridge SMUs Required    Caveats
=====
7.5.1     7.5.2     None                    None
=====
```

現在のイメージには、サポートされているアップグレードまたは特定のバージョンへのダウングレードバージョンのみを指定するアップグレードマトリックスがあります。現在のマトリックスのバージョンよりも新しいバージョンのISOのアップグレードパスを決定する場合は、新しいISOのアップグレードマトリックスに、サポートされているアップグレードまたはダウングレードパスが記載されています。

現在のバージョンからISOバージョンへのサポートされているリリースの表示

次に、ソフトウェアアップグレードパス、ダウングレードパス、および現在のバージョンからターゲットISOバージョンへのアップグレードの制限事項の例を示します。

```
Router#show install upgrade-matrix iso /harddisk:/ asr9k-goldenk9-x64-7.5.2-rev1.iso all
Fri Jul 29 10:20:10.961 IST
This may take a while ...
```

```
7.5.2 can be upgraded from and downgraded to the following releases:
```

```
=====
From      To        Bridge SMUs Required    Caveats
=====
7.0.2     7.5.2     r702.CSCvw57276        None
-----
7.5.1     7.5.2     None                    None
-----
7.4.2     7.5.2     None                    None
-----
7.4.1     7.5.2     None                    None
-----
```

7.5.2	7.0.2	None	None
7.5.2	6.5.3	None	- https://www.cisco.com/c/en/us/td/docs/routers/asr9000/software/asr9k-r7-4/system-security/configuration/guide/b-system-security-cg-asr9000-74x/configuring-aaa-services.html#id_128191
7.5.2	7.1.25	None	None
7.5.2	7.4.2	None	None
7.5.2	7.6.1	None	None
7.5.2	7.4.1	None	None
7.5.2	7.1.3	None	None
7.5.2	7.1.2	None	None
7.5.2	7.3.1	None	None
7.5.2	6.6.3	None	- https://www.cisco.com/c/en/us/td/docs/routers/asr9000/software/asr9k-r7-4/system-security/configuration/guide/b-system-security-cg-asr9000-74x/configuring-aaa-services.html#id_128191
7.5.2	7.3.2	None	None
7.1.25	7.5.2	None	None
7.1.3	7.5.2	None	None
7.1.2	7.5.2	None	None
7.6.1	7.5.2	None	None
6.5.3	7.5.2	r653.CSCvw57276	None
7.3.1	7.5.2	None	None
6.6.3	7.5.2	r663.CSCvw57276	None
7.3.2	7.5.2	None	None

実行中のバージョンから ISO バージョンへのサポートされているリリースの表示

次に、実行中のシステムとターゲット ISO イメージの両方のサポートマトリックスに従った現在のバージョンから ISO のバージョンへのソフトウェアのアップグレードに関する詳細を表示する例を示します。

```
Router#show install upgrade-matrix iso /harddisk:/asr9k-goldenk9-x64-7.5.2-rev1.iso
from-running
Fri Jul 29 10:21:31.957 IST
```

```
This may take a while ...
Upgrade from the current software [7.5.1] to 7.5.2 is supported
```

From	To	Bridge SMUs Required	Caveats
7.5.1	7.5.2	None	None

機能のアップグレード

機能のアップグレードとは、ルータに新機能とソフトウェアパッチを導入するプロセスです。パッケージをインストールして機能のアップグレードを実行します。ソフトウェアパッチのインストールは、ソフトウェアメンテナンスアップグレード（SMU）ファイルをインストールして実行します。

ルータにパッケージをインストールすると、そのパッケージに含まれる特定の機能がインストールされます。Cisco IOS XR ソフトウェアはさまざまなソフトウェアパッケージに分割されているため、ルータで実行する機能を選択できます。各パッケージには、ルーティングやセキュリティなど、特定のルータ機能のセットを実行するコンポーネントが含まれています。

たとえば、ルーティングパッケージのコンポーネントは、BGP や OSPF など、個別の RPM に分かれています。BGP は基本ソフトウェアバージョンの一部であり、必須 RPM なので削除できません。ただし、OSPF などの任意の RPM は、必要に応じて追加および削除できます。

パッケージの命名規則は <platform>-<pkg>-<pkg version>-<release version>.<architecture>.rpm です。

次に例を示します。

```
asr9k-9000v-nV-x64-1.0.0.0-r702.x86_64.rpm
```

install コマンドを使用して、パッケージと SMU をインストールします。インストールプロセスの詳細については、[パッケージのインストール（60 ページ）](#) を参照してください。



- (注)
- 十分なディスク容量があることを確認します。
 - fsck** コマンドを実行してファイルシステムのステータスを確認し、IOS XR が正常にアップグレードされていることを確認します。**fsck** コマンドは、システム管理パッケージをインストールする場合はシステム管理 EXEC モードで、XR パッケージをインストールする場合は XR EXEC モードで実行する必要があります。
 - すべての **install** コマンドは、システム管理 EXEC モードと XR EXEC モードの両方で適用できます。システム管理のインストール操作は XR EXEC モードで実行します。

XR VM とシステム管理 VM 用の個別のパッケージと SMU があります。それぞれをそのファイル名で識別できます。

たとえば、asr9k-px-7.9.1.CSCvu59908.pie は XR VM のパッケージの例です。asr9k-sysadmin-7.9.1.pie は、システム管理 VM に関連付けられます。

XR パッケージや SMU は XR VM からアクティブ化されますが、システム管理パッケージや SMU はシステム管理 VM からアクティブ化されます。

または、XR からシステム管理パッケージと SMU をアクティブ化または非アクティブ化することで、クロス VM 操作を実行できます。

最適化されたインストールイメージのサイズ

表 3: 機能の履歴 (表)

機能名	リリース情報	説明
最適化されたインストールイメージのサイズ	リリース 7.3.1	このリリースでは、ルータにインストールされている Cisco ISO イメージのサイズが約 300 MB 縮小されているため、ルータのディスク容量が解放され、パッケージのインストールにかかる時間が短縮されます。

Cisco IOS XR リリース 7.3.1 では、イメージ全体のサイズが約 300 MB 縮小され、最適化されたバージョンの mini-x ISO イメージが導入されています。この最適化された ISO により、インストール操作にかかる時間が短縮され、インストールリポジトリに保存されているファイルのディスク容量の使用率が削減されます。

この最適化された ISO は、次の方法を使用したルータの起動をサポートしています。

- iPXE ブート
- USB ブート
- システムのアップグレード
- In-Service Software Upgrade (ISSU)

ISSU の場合、下位バージョンからリリース 7.3.1 以降にアップグレードするには、実行中の下位バージョンにブリッジ SMU が必要です。一方、リリース 7.3.1 以降から下位バージョンへのロールバックには、ブリッジ SMU は必要ありません。SMU がインストールされているリリース 7.3.1 から古いバージョンにダウングレードする場合、SMU に依存関係はありません。ブリッジ SMU は必須の前提条件であり、リリース 7.3.1 以降へのアップグレードを実行する前に、実行中のバージョンにインストールする必要があります。

次に、下位の実行中のバージョンにインストールする必要があるホスト、XR、およびシステム管理用のブリッジ SMU を示します。たとえば、リリース 7.1.1 の場合は次のようになります。

- `asr9k-iosxr-infra-64-3.0.0.2-r711.CSCvq46421.x86_64.rpm`

- asr9k-iosxr-os-64-3.0.0.2-r711.CSCvq46421.x86_64.rpm
- asr9k-sysadmin-system-7.1.1-r711.CSCvq46421.x86_64.rpm

ブリッジ SMU は、Download Software Center にある特定のリリースの Cisco IOS XR ソフトウェアの tar バンドルに含まれています。

インストール操作に関連するコマンドでは、common という用語を含む新しいパッケージが表示されます。共通パッケージは、NETCONF YANG モードを使用して実行した場合、install コマンドの出力にも表示されます。



(注) **show install active summary** および **show install committed summary** コマンドの common 用語はリストされていません。

共通パッケージは、システム管理 VM の /misc/disk1/tftpboot にあります。

common という用語は、システム管理 VM および XR VM の **show install repository** または **show install repository all** コマンドの出力にも表示されます。

次に、システム管理 VM からの出力例を示します。

```
sysadmin-vm:0_RSP0# show install repository
Sat May 2 16:33:25.200 UTC+00:00
Admin repository
-----
asr9k-common-7.3.1
asr9k-goldenk9-x64-7.3.1-supersetOptInstallSmu
asr9k-mini-x64-7.3.1
asr9k-sysadmin-7.3.1
asr9k-sysadmin-hostos-7.3.1-r731.admin.x86_64
asr9k-sysadmin-hostos-7.3.1-r731.host.x86_64
asr9k-sysadmin-hostos-7.3.1-r731.CSCho77777.admin.x86_64
asr9k-sysadmin-hostos-7.3.1-r731.CSCho77777.host.x86_64
asr9k-sysadmin-hostos-7.3.1-r731.admin.x86_64
asr9k-sysadmin-hostos-7.3.1-r731.host.x86_64
asr9k-sysadmin-hostos-7.3.1-r731.CSCho77777.admin.x86_64
asr9k-sysadmin-hostos-7.3.1-r731.CSCho77777.host.x86_64
asr9k-sysadmin-shared-7.3.1-r731.CSCcv33333.x86_64
asr9k-sysadmin-shared-7.3.1-r731.CSCcv33333.x86_64
asr9k-sysadmin-system-7.3.1-r731.CSCcv11111.x86_64
asr9k-sysadmin-system-7.3.1-r731.CSCcv22222.x86_64
----- Truncated for Brevity -----
```

次に、XR VM からの出力例を示します。

```
Router#show install repository
Sat May 2 13:55:59.996 UTC
26 package(s) in XR repository:
asr9k-iosxr-infra-64-1.0.0.1-r731.CSCxr22222.x86_64
asr9k-iosxr-infra-64-1.0.0.2-r731.CSCxr44444.x86_64
asr9k-k9sec-x64-2.1.0.0-r731.x86_64
asr9k-optic-x64-1.0.0.0-r731.x86_64
asr9k-mgbl-x64-2.0.0.0-r731.x86_64
asr9k-mpls-te-rsvp-x64-2.1.0.0-r731.x86_64
asr9k-common-7.3.1
asr9k-goldenk9-x64-7.3.1-supersetOptInstallSmu
```

```
asr9k-bng-x64-1.0.0.0-r731.x86_64
asr9k-mini-x64-7.3.1
asr9k-mpls-x64-2.0.0.0-r731.x86_64
asr9k-li-x64-1.1.0.0-r73104I.x86_64
----- Truncated for Brevity -----
```

コマンド出力の `asr9k-common-7.3.1` は、イメージサイズを縮小するために最適化されたパッケージを表します。

準備済みパッケージのインストール

システムアップグレードまたは機能アップグレードは、ISO イメージファイル、パッケージ、および SMU をアクティブ化することで実行します。アクティベーション前にこれらのインストール可能なファイルを準備することができます。準備フェーズでは、アクティベーション前のチェックが実行され、インストール可能なファイルのコンポーネントがルータ設定にロードされます。準備プロセスはバックグラウンドで実行されるため、その間もルータをフルに利用できます。準備フェーズが完了したら、すべての準備済みファイルを即座にアクティブ化できます。アクティベーション前の準備には、次の利点があります。

- インストール可能なファイルが破損していると、準備プロセスは失敗します。これによって問題が早期に警告されます。破損したファイルが直接アクティブ化されると、ルータの誤動作を招く可能性があります。
- システムアップグレード用の ISO イメージを直接アクティブ化するには時間がかかり、その間にルータを使用できなくなります。ただし、アクティベーション前にイメージを準備すると、準備プロセスが非同期で実行されるだけでなく、準備済みのイメージを後でアクティブ化するとき、アクティベーションプロセスにかかる時間も短縮されます。その結果、ルータのダウンタイムが大幅に削減されます。
- 正常な操作に必要なディスク容量チェックが実行されるため、ディスク容量の不足が定量化され、ファイルシステムの領域を解放するための可能な代替手段が提供されます。
- パッケージの互換性チェックが実行されるため、必要なすべてのインストールパッケージが使用可能になります。パッケージ互換性チェックエラーの場合、パッケージとバージョンの詳細がログに記録されます。

システムのアップグレードおよびパッケージのインストールに準備動作を利用するには、次のタスクを実行します。



- (注) システム管理パッケージまたは XR パッケージのどちらをインストールするかによって、それぞれ `install` モードまたは `install` モードで `install` コマンドを実行します。すべての `install` コマンドは両方のモードで使用できます。システム管理のインストール動作は XR モードで実行できます。

ステップ 1 必要な ISO イメージおよびパッケージをリポジトリに追加します。

詳細については、[パッケージのインストール \(60 ページ\)](#) を参照してください。

ステップ2 show install repository

例：

```
RP/0/RSP0/cpu 0: router#show install repository
```

必要なインストール可能ファイルがリポジトリ内にあることを確認するには、この手順を実行します。パッケージは「install add」動作の完了後にのみ表示されます。

ステップ3 show install request

例：

```
RP/0/RSP0/cpu 0: router#show install request
```

(任意) 追加動作の動作 ID とステータスを表示します。動作 ID は、後で **activate** コマンドを実行する際に使用できます。

```
Install operation 8 is still in progress
```

ステップ4 次のいずれかを実行します。

- **install prepare** *package_name*
- **install prepare id** *operation_id*

例：

準備プロセスが開始されます。この動作は非同期モードで実行されます。**install prepare** コマンドはバックグラウンドで実行され、EXEC プロンプトは最短で返されます。

動作 ID を使用すると、指定した動作に追加されたすべてのパッケージの準備がまとめて行われます。たとえば 5 つのパッケージが動作 8 に追加されている場合、**install prepare id 8** を実行すると、5 つのパッケージがすべてまとめて準備されます。パッケージを個別に準備する必要はありません。

ステップ5 show install prepare

例：

```
RP/0/RSP0/cpu 0: router#show install prepare
```

準備済みのパッケージを表示します。この結果で、必要なすべてのパッケージが準備されていることを確認します。

ステップ6 install activate

例：

```
RP/0/RSP0/cpu 0: router#install activate
```

準備の完了したすべてのパッケージをまとめてアクティブ化し、ルータでパッケージ設定をアクティブにします。

(注) CLI でパッケージ名または動作 ID を指定しないでください。

SMU によっては、アクティベーション時にルータの手動リロードが必要な場合があります。このような SMU をアクティブ化すると、リロードを実行するための警告メッセージが表示されます。SMU のコンポーネントは、リロードの完了後にのみアクティブ化されます。**install activate** コマンドの完了後すぐにルータのリロードを実行します。

ステップ7 show install active

例：

```
RP/0/RSP0/cpu 0: router#show install active
```

アクティブなパッケージを表示します。

この結果で、すべての RP と LC でイメージおよびパッケージの同じバージョンがアクティブになっていることを確認します。

ステップ 8 install commit

例：

```
RP/0/RSP0/cpu 0: router#install commit
```

パッケージのインストール：関連コマンド

関連コマンド	目的
show install log	インストールプロセスのログ情報を表示します。これはインストールが失敗した場合のトラブルシューティングに使用できます。
show install package	リポジトリに追加されたパッケージの詳細を表示します。このコマンドは、パッケージの個々のコンポーネントを識別する際に使用します。
install prepare clean	準備動作をクリアし、すべてのパッケージを準備済み状態から削除します。

次のタスク

- システムのアップグレードの実行後、システム管理 モードから **upgrade hw-module location all fpd all** コマンドを使用して FPD をアップグレードします。FPD アップグレードプロセスの進捗状況は、システム管理 モードで **show hw-module fpd** コマンドを使用してモニターできます。FPD アップグレードが完了したら、ルータをリロードします。
- **install verify packages** コマンドを使用してインストールを確認します。
- インストールによってルータに問題が発生した場合は、該当するパッケージまたは SMU をアンインストールしてください。[パッケージのアンインストール](#)を参照してください。



(注) ISO イメージはアンインストールできません。ただし、旧バージョンの ISO をインストールすることでシステムダウングレードを実行することができます。

パッケージのインストール

システムをアップグレードするか、パッチをインストールするには、このタスクを完了します。システムアップグレードは ISO イメージファイルを使用して行いますが、パッチインストールの場合はパッケージおよび SMU を使用します。アップグレード操作には、ミニ ISO とともに SMU も含めることができます。

.rpm ファイルをインストールする際もこのタスクを使用します。rpm ファイルには、1つのファイルに統合された複数のパッケージと SMU が含まれています。カードタイプにかかわらず、パッケージ形式によってコンポーネントごとに 1つの RPM が定義されます。



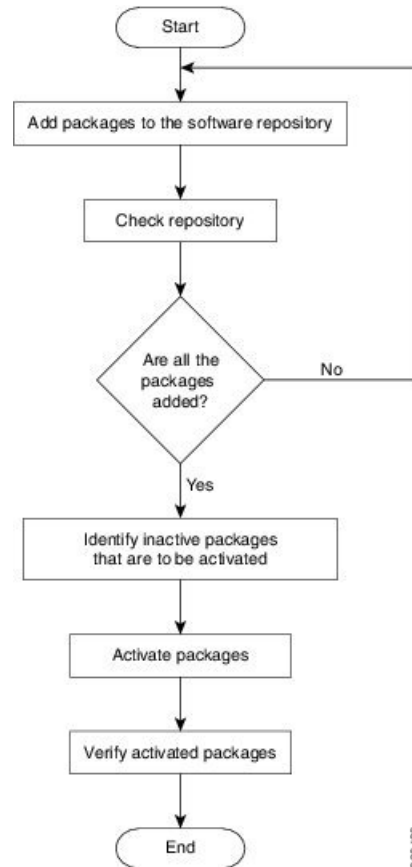
- (注)
- 十分なディスク容量があることを確認します。
 - **fsck** コマンドを実行してファイルシステムのステータスを確認し、IOS XR が正常にアップグレードされていることを確認します。**fsck** コマンドは、システム管理パッケージをインストールする場合はシステム管理 EXEC モードで、XR パッケージをインストールする場合は XR EXEC モードでを実行する必要があります。
 - すべての **install** コマンドは、システム管理 EXEC モードと XR EXEC モードの両方で適用できます。システム管理のインストール操作は XR EXEC モードで実行します。



- (注)
- システムアップグレードは、XR EXEC モードからのみサポートされています。
 - システム管理パッケージはシステム管理 EXEC モードおよび XR EXEC モードで **install** コマンドを使用して実行できますが、XR パッケージは XR EXEC モードで **install** コマンドを使用してのみ実行できます。すべての **install** コマンドは両方のモードで使用できます。
 - システム管理 SMU はシステム管理 EXEC モードおよび XR EXEC モードでインストールできますが、XR SMU は XR EXEC モードでのみインストールできます。
 - IPv6 を介したインストール操作はサポートされていません。

パッケージをインストールするためのワークフローを次のフローチャートに示します。

図 3: パッケージインストールのワークフロー



始める前に

- 管理ポートを設定して接続します。インストール可能なファイルには管理ポートからアクセスできます。
- インストールするパッケージを、ルータのハードディスク、またはルータがアクセスできるネットワーク サーバーにコピーします。

ステップ 1 次のいずれかを実行します。

- **install add source** <http or shhttp transfer protocol>/package_path/ filename1 filename2 ...
- **install add source** <tftp transfer protocol>/package_path/ filename1 filename2 ...
- **install add source** <ftp or sftp transfer protocol>/user@server:/package_path/ filename1 filename2 ...

例：

```
RP/0/RSP0/cpu 0: router#install add source /harddisk:/ asr9k-mpls-x64-2.1.0.0-r731.x86_64.rpm
asr9k-mpls-x64-2.1.0.0-r732.x86_64.rpm
```

または

■ パッケージのインストール

```
RP/0/RSP0/cpu 0: router#install add source sftp://root@8.33.5.15:/auto/ncs/package/
asr9k-mcast-1.0.0.0-731.x86_64.rpm asr9k-iosxr-mpls-1.0.0.0-732.x86_64.rpm
```

または

```
RP/0/RSP0/cpu 0: router#install add source /harddisk:/
asr9k-mpls-x64-2.1.0.0-<release-number>.x86_64.rpm asr9k-mpls-x64-2.1.0.0-<release-number>.x86_64.rpm
```

または

```
RP/0/RSP0/cpu 0: router#install add source sftp://root@8.33.5.15:/auto/ncs/package/
asr9k-mcast-1.0.0.0-<release-number>.x86_64.rpm asr9k-iosxr-mpls-1.0.0.0-<release-number>.x86_64.rpm
```

(注) `package_path` と `filename` の間にはスペースが必要です。

パッケージからソフトウェアファイルが展開され、検証されてからソフトウェアリポジトリに追加されます。追加するファイルのサイズによっては、この処理に時間がかかる場合があります。動作は非同期モードで実行されます。**install add** コマンドはバックグラウンドで実行され、すべてのファイルが展開されると EXEC プロンプトが返されます。

(注) XR VM とシステム管理 VM のリポジトリは異なります。ルーティングパッケージは XR VM リポジトリに、システム管理パッケージはシステム管理 VM リポジトリに自動的に追加されます。

ステップ 2 show install request

例：

```
RP/0/RSP0/cpu 0: router#show install request
```

(任意) 追加動作の動作 ID とステータスを表示します。動作 ID は、後で **activate** コマンドを実行する際に使用できます。

```
Install operation 8 is still in progress
```

ステップ 3 show install repository

例：

```
RP/0/RSP0/cpu 0: router#show install repository
```

リポジトリに追加されるパッケージを表示します。パッケージは `install add` 動作の完了後にのみ表示されます。

ステップ 4 show install inactive

例：

```
RP/0/RSP0/cpu 0: router#show install inactive
```

リポジトリ内に存在する非アクティブなパッケージを表示します。アクティブ化できるのは非アクティブなパッケージだけです。

ステップ 5 次のいずれかを実行します。

- **install activate** `package_name`
- **install activate id** `operation_id`

例：

`operation_id` は **install add** 操作の ID です。 [パッケージのインストール \(60 ページ\)](#) の [ステップステップ 2 \(62 ページ\)](#) を参照してください。このコマンドは、システム管理モードからも実行できます。パッケージ設定がルータでアクティブになります。その結果、新機能とソフトウェア修正が有効になります。この操作は、デフォルトの非同期モードで実行されます。**install activate** コマンドはバックグラウンドで実行され、EXEC プロンプトが返されます。

同期モードを使用するか、CLI から `[sync]` オプションを選択して、アクティブ化操作を実行できます。

動作 ID を使用すると、指定した動作に追加されたすべてのパッケージがまとめてアクティブ化されます。たとえば、5 つのパッケージが動作 ID 8 に追加されている場合、**install activate id 8** を実行すると、5 つのパッケージがすべてまとめてアクティブ化されます。パッケージを個別にアクティブ化する必要はありません。

アクティベーションは瞬時には完了せず、ある程度の時間がかかります。アクティブ化が完了すると、システムが自動的にリロードします。再起動 SMU アクティベーションの場合、SMU の影響を受けるプロセスが再起動されると、SMU が有効になります。

SMU が XR VM とシステム管理 VM の両方と依存関係がある場合は、両方の VM で SMU をアクティブ化した後でリロードを実行すると、両方で同時に反映されます。ルータをリロードするには、システム管理 EXEC モードから **hw-module location all reload** コマンドを使用します。

ステップ 6 show install active

例：

```
RP/0/RSP0/cpu 0: router#show install active
```

アクティブなパッケージを表示します。

この結果で、すべての RP と LC でイメージおよびパッケージの同じバージョンがアクティブになっていることを確認します。

表 4: 例：パッケージのインストール：関連コマンド

関連コマンド	目的
show install log	インストールプロセスのログ情報を表示します。これはインストールが失敗した場合のトラブルシューティングに使用できます。
show install package	リポジトリに追加されたパッケージの詳細を表示します。このコマンドは、パッケージの個々のコンポーネントを識別する際に使用します。
install prepare	アクティベーションの準備として、非アクティブなパッケージに対してアクティベーション前のチェックを実行します。
show install prepare	準備が完了してアクティベーション可能になったパッケージのリストを表示します。

ステップ 7 install commit

例：

```
RP/0/RSP0/cpu 0: router#install commit
```

ホスト、XR、およびシステム管理で新たにアクティブになったソフトウェアをコミットします。

- (注) マルチ SDR モードでは、**install commit sdr** を使用して、CLI がトリガーされている場所から SDR のみをコミットできます。詳細については、「[Secure Domain Router Commands](#)」を参照してください。

次のタスク

- システムのアップグレードの実行後、システム管理 モードから **upgrade hw-module location all fpd all** コマンドを使用して FPD をアップグレードします。FPD アップグレードプロセスの進捗状況は、システム管理 モードで **show hw-module fpd** コマンドを使用してモニターできます。FPD アップグレードが完了したら、ルータをリロードします。
- **install verify packages** コマンドを使用してインストールを確認します。
- インストールによってルータに問題が発生した場合は、該当するパッケージまたは SMU をアンインストールしてください。[パッケージのアンインストール \(64 ページ\)](#) を参照してください。

パッケージのアンインストール

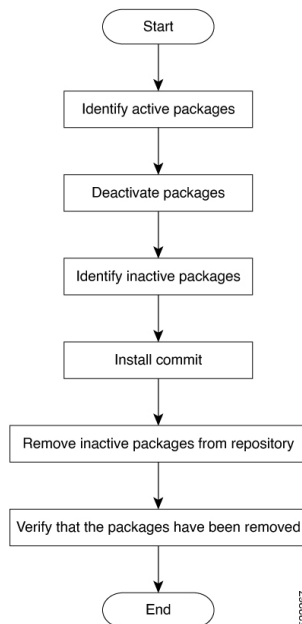
パッケージをアンインストールするには、次のタスクを実行します。アンインストールしたパッケージに含まれるすべてのルータ機能は非アクティブ化されます。XR VM で追加したパッケージは、システム管理 VM からはアンインストールできません。ただし、クロス VM 操作では、XR からシステム管理パッケージを非アクティブ化できます。



- (注) インストール済みの ISO イメージはアンインストールできません。また、ホスト、XR VM、およびシステム管理 VM でサードパーティ製 SMU をインストールするカーネル SMU もアンインストールできません。ただし、ISO イメージまたはカーネル SMU を新たにインストールすると既存のインストールが上書きされます。

パッケージをアンインストールするためのワークフローを次のフローチャートに示します。

図 4: パッケージアンインストールのワークフロー



このタスクでは、XR VM パッケージをアンインストールします。システム管理パッケージをアンインストールする場合は、同じコマンドを `mode` で実行します。

ステップ 1 show install active

例 :

```
RP/0/RSP0/cpu 0: router#show install active
```

アクティブなパッケージを表示します。非アクティブ化できるのはアクティブなパッケージだけです。

ステップ 2 次のいずれかを実行します。

- `install deactivate package_name`
- `install deactivate id operation_id`

例 :

`operation_id` は `install add` 動作の ID です。パッケージに関連するすべての機能およびソフトウェアパッチが非アクティブ化されます。複数のパッケージ名を指定して同時に非アクティブ化できます。

動作 ID を使用すると、指定した動作に追加されたすべてのパッケージがまとめて非アクティブ化されません。パッケージを個別に非アクティブ化する必要はありません。 `install add` 動作（非アクティブ化で使った ID の動作）の一部として追加されたシステム管理パッケージがある場合、これらも非アクティブ化されます。

ステップ 3 show install inactive

例 :

```
RP/0/RSP0/cpu 0: router#show install inactive
```

非アクティブ化済みのパッケージは、非アクティブなパッケージとして表示されるようになります。非アクティブなパッケージのみリポジトリから削除できます。

ステップ 4 `install commit`

ステップ 5 `install remove package_name`

例：

非アクティブなパッケージがリポジトリから削除されます。

指定した動作 ID に追加されているすべてのパッケージを削除するには、`id operation-id` キーワードおよび引数を指定して `install remove` コマンドを使用します。

また、`install remove inactive all` を使用して、XR およびシステム管理からすべての非アクティブなパッケージを削除できます。

ステップ 6 `show install repository`

例：

```
RP/0/RSP0/cpu 0: router#show install repository
```

リポジトリ内の使用可能なパッケージを表示します。削除されたパッケージは結果に表示されなくなります。

次のタスク

必要なパッケージをインストールします。。

プラットフォームでサポートされている機能の表示

表 5: 機能の履歴 (表)

機能名	リリース情報	説明
プラットフォームでサポートされている機能の表示	リリース 7.5.2	この機能には、ルータのリリースでサポートされている機能とサポートされていない機能のリストが表示されます。この機能を使用すると、サポート情報用の注釈付き機能を使用して、ネットワーク設定をより適切に計画できます。この機能では、 <code>show features</code> コマンドが導入されています。

この機能では、「Is feature X supported on my router?」という質問に対する回答を得られません。

使用しているルータのリリースで機能がサポートされているかどうかを確認できます。サポート情報は、リリースおよびプラットフォーム固有のデータ（ルータに存在するプラットフォームバリエーション、RP、LC など）に基づいています。



- (注) Cisco IOS XR ソフトウェアリリース 7.5.2 では、アクセス制御リスト (ACL) 機能のみがサポートされています。

サポートされているリリースがルータにインストールされている場合、機能情報を決定する機能はデフォルトで有効になっています。

サポートされている機能と機能のリストを表示するには、**show features** コマンドを使用します。機能は、サポート情報の表記とともにツリー構造で表示されます。たとえば、ACL では、多数のアクセス制御エレメント (ACE) に対応するために圧縮を使用する機能がサポートされていますが、IPv6 ACL BNG には、Cisco IOS XR ソフトウェアリリース 7.5.2 のサポートデータがありません。機能に関するこのサポート情報は、ツリー構造の次のキーで表されます。

キー	機能サポート情報	説明
X	Unsupported	機能は、当該リリースのプラットフォームではサポートされていません。
-	サポート対象	機能は、当該リリースのプラットフォームでサポートされています。
?	サポートは不明	機能のサポートについては、当該リリースのプラットフォームでは不明です。機能のオプションパッケージがルータにインストールされていないことが原因である可能性があります。
*	サポートデータがありません	機能のサポートは、当該リリースのプラットフォームでは使用できません。機能がルータに存在しないラインカードに固有であることが原因である可能性があります。

サポート対象機能のリストの表示

この例では、ルータでサポートされている機能が表示されます。



- (注) Cisco IOS XR ソフトウェアリリース 7.5.2 では、アクセス制御リスト (ACL) の機能のみがサポートされています。

```
Router#show features
Fri Jun 3 19:16:58.298 UTC
Key:
X - Unsupported
- - Supported
? - Support unknown (optional package not installed)
* - Support data not available

[-] Cisco IOS XR
```

プラットフォームでサポートされている機能の表示

```

|--[-] XR Protocols
| |--[-] XR Base Protocols
| | |--[-] Services
| | | |--[-] Access Control List (ACL)
| | | | |--[-] IPv6 ACL Support
| | | | | |--[*] IPv6 ACL ABF Track
| | | | | |--[*] IPv6 ACL BNG
| | | | | |--[*] IPv6 ACL Chaining (Meta ACL)
| | | | | |--[-] IPv6 ACL Common ACL
| | | | | |--[-] IPv6 ACL Compression
| | | | | |--[*] IPv6 ACL Default ABF
| | | | | |--[*] IPv6 ACL Fragment
| | | | | |--[-] IPv6 ACL ICMP Off
| | | | | |--[-] IPv6 ACL ICMP Protocol
| | | | | |--[-] IPv6 ACL Interface Statistics
| | | | | |--[-] IPv6 ACL Log Rate
| | | | | |--[-] IPv6 ACL Log Threshold
| | | | | |--[-] IPv6 ACL Logging
| | | | | |--[-] IPv6 ACL MIB
| | | | | |--[-] IPv6 ACL Object Groups (Scale)
| | | | | |--[-] IPv6 ACL Police
| | | | | |--[-] IPv6 ACL Priority
| | | | | |--[*] IPv6 ACL Protocol Range
| | | | | |--[-] IPv6 ACL Set Qos-Group
| | | | | |--[-] IPv6 ACL Set TTL
| | | | | |--[-] IPv6 ACL TCP Flags
| | | | | |--[-] IPv6 ACL TTL Match
| | | | | |--[-] IPv6 ACL UDF
| | | | |--[-] ES-ACL Support (L2 ACL)
| | | |--[-] IPv4 ACL Support
| | | | |--[-] IPv4 ACL Set Qos-group
| | | | |--[*] IPv4 ACL ABF Track
| | | | |--[*] IPv4 ACL BNG
| | | | |--[*] IPv4 ACL Chaining (Meta ACL)
| | | | |--[-] IPv4 ACL Common ACL
| | | | |--[-] IPv4 ACL Compression
| | | | |--[*] IPv4 ACL Default ABF
| | | | |--[*] IPv4 ACL Fragment
| | | | |--[-] IPv4 ACL Fragment Flags
| | | | |--[-] IPv4 ACL ICMP Off
| | | | |--[-] IPv4 ACL ICMP Protocol
| | | | |--[-] IPv4 ACL Interface Statistics
| | | | |--[-] IPv4 ACL Log Rate
| | | | |--[-] IPv4 ACL Log Threshold
| | | | |--[-] IPv4 ACL Logging
| | | | |--[-] IPv4 ACL MIB
| | | | |--[-] IPv4 ACL Object Groups (Scale)
| | | | |--[-] IPv4 ACL Police
| | | | |--[-] IPv4 ACL Priority
| | | | |--[*] IPv4 ACL Protocol Range
| | | | |--[-] IPv4 ACL Set TTL
| | | | |--[-] IPv4 ACL TCP Flags
| | | | |--[-] IPv4 ACL TTL
| | | | |--[-] IPv4 ACL UDF
| | | |--[-] IPv4 Prefix-List
| | |--[-] IPv6 Prefix-List

```

サポートされている ACL 機能のリストの表示

この例では、ルータの ACL 機能が表示されます。

```
Router#show features acl
Fri Jun 3 19:17:31.635 UTC
Key:
X - Unsupported
- - Supported
? - Support unknown (optional package not installed)
* - Support data not available

[-] Access Control List (ACL)
|--[-] IPv6 ACL Support
| |--[*] IPv6 ACL ABF Track
| |--[*] IPv6 ACL BNG
| |--[*] IPv6 ACL Chaining (Meta ACL)
| |--[-] IPv6 ACL Common ACL
| |--[-] IPv6 ACL Compression
| |--[*] IPv6 ACL Default ABF
| |--[*] IPv6 ACL Fragment
| |--[-] IPv6 ACL ICMP Off
| |--[-] IPv6 ACL ICMP Protocol
| |--[-] IPv6 ACL Interface Statistics
| |--[-] IPv6 ACL Log Rate
| |--[-] IPv6 ACL Log Threshold
| |--[-] IPv6 ACL Logging
| |--[-] IPv6 ACL MIB
| |--[-] IPv6 ACL Object Groups (Scale)
| |--[-] IPv6 ACL Police
| |--[-] IPv6 ACL Priority
| |--[*] IPv6 ACL Protocol Range
| |--[-] IPv6 ACL Set Qos-Group
| |--[-] IPv6 ACL Set TTL
| |--[-] IPv6 ACL TCP Flags
| |--[-] IPv6 ACL TTL Match
| |--[-] IPv6 ACL UDF
|--[-] ES-ACL Support (L2 ACL)
|--[-] IPv4 ACL Support
| |--[-] IPv4 ACL Set Qos-group
| |--[*] IPv4 ACL ABF Track
| |--[*] IPv4 ACL BNG
| |--[*] IPv4 ACL Chaining (Meta ACL)
| |--[-] IPv4 ACL Common ACL
| |--[-] IPv4 ACL Compression
| |--[*] IPv4 ACL Default ABF
| |--[*] IPv4 ACL Fragment
| |--[-] IPv4 ACL Fragment Flags
| |--[-] IPv4 ACL ICMP Off
| |--[-] IPv4 ACL ICMP Protocol
| |--[-] IPv4 ACL Interface Statistics
| |--[-] IPv4 ACL Log Rate
| |--[-] IPv4 ACL Log Threshold
| |--[-] IPv4 ACL Logging
| |--[-] IPv4 ACL MIB
| |--[-] IPv4 ACL Object Groups (Scale)
| |--[-] IPv4 ACL Police
| |--[-] IPv4 ACL Priority
| |--[*] IPv4 ACL Protocol Range
| |--[-] IPv4 ACL Set TTL
| |--[-] IPv4 ACL TCP Flags
| |--[-] IPv4 ACL TTL
| |--[-] IPv4 ACL UDF
|--[-] IPv4 Prefix-List
|--[-] IPv6 Prefix-List
```

特定の RP でサポートされている ACL 機能のリストの表示

この例では、RP ロケーション 0/RP0/CPU0 の ACL 機能が表示されます。

```
Router#show features acl detail location 0/RP0/CPU0
Fri Jun 3 19:15:49.889 UTC
Key:
X - Unsupported
- - Supported
? - Support unknown (optional package not installed)
* - Support data not available

[-] Access Control List (ACL)
Cisco provides basic traffic filtering capabilities with access control
lists (also referred to as access lists). User can configure access
control lists (ACLs) for all routed network protocols to filter protocol
packets when these packets pass through a device. User can configure
access lists on your device to control access to a network, access lists
can prevent certain traffic from entering or exiting a network.

|--[-] IPv6 ACL Support
|
| IPv6 based ACL is a list of source IPv6 addresses that use Layer 3 or
| Layer 4 information to permit or deny access to traffic. IPv6 router
| ACLs apply only to IPv6 packets that are routed.. A filter contains the
| rules to match the packet matches, the rule also stipulates if the
| packet should be permitted or denied.
|
|--[*] IPv6 ACL ABF Track
|
| IPv6 ACL ABF Track allows the user to configure a rule with track as
| nextthop inside the ACL rule . ACL Based Forwarding (ABF) denotes the
| ability to forward packets to another next hop router based on the
| criteria defined in the rule. Track takes precedence over VRF and
| IP, if present in the nextthop
|
|--[*] IPv6 ACL BNG
|
| IPv6 ACL BNG is an ACL subscriber BNG feature. It allows the use of
| ACL on dynamic template.
|
|--[*] IPv6 ACL Chaining (Meta ACL)
|
| IPv6 ACL Chaining (Meta ACL) allows the user to apply more than one
| ACL on the interface. is known as Meta ACL or ACL chaining.
|
|--[-] IPv6 ACL Common ACL
|
| IPv6 ACL Common allows the user to apply the ACL on the interface
| using the common keyword. Using this feature the ACL won't be
| applied to the specific interface but it will be common to th entire
| NPU to which the interface belongs.
|
|--[-] IPv6 ACL Compression
|
| IPv6 ACL Compression allows the user to apply the ACL on the
| interface using a compression level. This helps in reducing the
| hardware resources needed to program the ACL.
|
|--[*] IPv6 ACL Default ABF
|
| IPv6 ACL Default ABF allows the user to configure a rule with
| default nextthop inside the ACL rule . ACL Based Forwarding (ABF)
| denotes the ability to forward packets to another next hop router
| based on the criteria defined in the rule
|
|--[*] IPv6 ACL Fragment
|
| IPv6 ACL Fragment allows the user to configure a rule with fragment
| inside the ACL rule and use it as a match criteria to filter traffic.
|
|--[-] IPv6 ACL ICMP Off
|
| IPv6 ACL ICMP Off allows the user to not genearte the ICMP error
| message on a deny action. When configured it will not send the
| packet to FIB to generate ICMP error message.
----- Truncated for Brevity -----
```

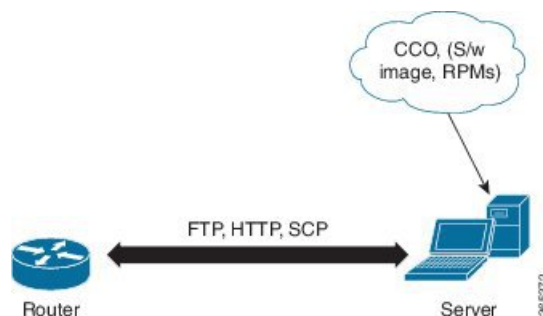


第 7 章

自動依存関係管理

フレキシブルパッケージでは、自動依存関係管理がサポートされます。RPM の更新中に、関連するすべての依存パッケージをシステムが自動的に特定し、それらを更新します。

図 5: インストールフロー（基本ソフトウェア、RPM、および SMU）



このリリースまでは、ソフトウェアイメージと必要な RPM をネットワークサーバー（リポジトリ）の CCO からダウンロードし、**install add** コマンドと **install activate** コマンドを使用してルータに追加し、アクティブにしていました。さらに、関連する依存 RPM を手動で特定し、それらを追加してアクティブにしていました。

自動依存関係管理を使用した場合は、依存 RPM を特定し、個別に追加およびアクティブにする必要がありません。新しいインストールコマンドを実行すると、依存 RPM は自動的にインストールされます。

install source コマンドは、パッケージを追加してアクティブにします。**install replace** コマンドは、特定のゴールデン ISO (GISO) にパッケージを追加してアクティブにします。



- (注) 1. Cisco IOS XR バージョン 6.1.1 では、サードパーティ製 SMU (**install source** コマンド) は、自動依存関係管理に含まれません。サードパーティ製 SMU は、他のインストール手順 (IOS XR の SMU および RPM、または管理コンテナのインストール) とは独立して、個別にインストールする必要があります。

これ以降で説明する内容は、次のとおりです。

- [RPM と SMU の更新 \(72 ページ\)](#)
- [基本ソフトウェア バージョンのアップグレード \(73 ページ\)](#)
- [RPM のダウングレード \(74 ページ\)](#)

RPM と SMU の更新

RPM には特定の不具合に対する修正が含まれており、その修正でシステムを更新する必要があります。RPM および SMU を新しいバージョンに更新するには、**install source** コマンドを使用します。特定の RPM に対してこのコマンドが発行されると、ルータによりリポジトリとの間で通信が行われ、RPM がダウンロードされてアクティブにされます。依存関係にある RPM がリポジトリにある場合、ルータによってその RPM も特定されてインストールされます。

install source コマンドの構文は次のとおりです。

install source リポジトリ **[rpm]**

install source コマンドは、次の 4 つの方法で実行できます。

- パッケージ名を指定しない。

パッケージ名を指定しないと、すべてのインストール済みパッケージがコマンドによって最新の SMU で更新されます。

```
install source [repository]
```

- パッケージ名を指定する。

パッケージ名を指定すると、そのパッケージがコマンドによってインストールされ、依存関係とともにそのパッケージの最新の SMU で更新されます。パッケージがすでにインストールされている場合、そのパッケージの SMU だけがインストールされます (すでにインストールされている SMU は、スキップされます)。

```
install source [repository] asr9k-mpls.rpm
```

- パッケージ名とバージョン番号を指定する。

パッケージの特定のバージョンをインストールする必要がある場合、完全なパッケージ名を指定します。このパッケージは、リポジトリにあるパッケージの最新の SMU とともにインストールされます。

```
install source [repository] asr9k-mpls-1.0.2.0-r710.x86_64.rpm
```

- SMU を指定する。

SMU を指定すると、その SMU は依存関係にある SMU とともにダウンロードおよびインストールされます。

```
install source [repository] asr9k-mpls-1.0.2.1-r611.CSCub12345.x86_64.rpm
```

基本ソフトウェアバージョンのアップグレード

基本ソフトウェアは、新しいバージョンが利用可能になった場合に、新しいバージョンにアップグレードできます。基本ソフトウェアを最新バージョンにアップグレードするには、**install source** コマンドを使用します。ベースバージョンをアップグレードすると、ルータで現在利用可能な RPM もアップグレードされます。



(注) SMU は、このプロセスの一部としてアップグレードされません。

install source コマンドの構文は次のとおりです。

```
install source repository
```



(注) データポート上の VRF および TPA はサポートされません。デフォルト以外の VRF インターフェイスを介してしかサーバーにアクセスできない場合、ファイルは、ftp、sftp、scp、http、または https プロトコルを使用してすでに取得しておく必要があります。



(注) TPA を実装しているため、デフォルトルート (0.0.0.0) は Linux にコピーできません。

install source コマンドを使用するのは次の場合です。

- バージョン番号を指定する

基本ソフトウェア (.mini) は、特定のバージョンにアップグレードされます。すべてのインストール済み RPM も、同じリリースバージョンにアップグレードされます。

```
install source [repository] version <version>  
asr9k-mini-x64-<version>.iso
```

次の例を参考にしてください。

```
install source repository version 7.0.1 asr9k-mini-x64-7.0.1.iso
```

必要なリリースの .mini ファイルと RPM を自動的に取得して、アップグレードを続行することもできます。

```
install source repository asr9k-mini-x64-7.0.1.iso
```

- RPM のバージョン番号を指定する

システムアップグレードを実行する場合、ユーザーはオプションの RPM (基本ソフトウェアのバージョンのリソースとは異なるリリース) を選択できます。つまり、RPM を指定できます。

```
install source repository version 7.0.1  
asr9k-mpls-1.0.2.0-r701.x86_64.rpm
```

RPM のダウングレード

RPM は、アクティブ化後にダウングレードできます。RPM には次のタイプがあります。

- **hostos RPM** : RPM の名前に `hostos` が含まれています。

次に例を示します。

- `<platform>-sysadmin-hostos-6.5.1-r651.CSChu77777.host.arm`
- `<platform>-sysadmin-hostos-6.5.1-r651.CSChu77777.admin.arm`
- `<platform>-sysadmin-hostos-6.5.1-r651.CSChu77777.host.x86_64`
- `<platform>-sysadmin-hostos-6.5.1-r651.CSChu77777.admin.x86_64`

- **hostos 以外の RPM** : RPM の名前に `hostos` が含まれていません。

次に例を示します。

- `<platform>-sysadmin-system-6.5.1-r651.CSCvc12346`

RPM を非アクティブにするには、次の手順を実行します。

- **hostos RPM のダウングレード**

- シナリオ 1 : アクティブバージョンの 09 からバージョン 06 にダウングレードするには、次の手順を実行します。

1. バージョン 06 の `hostos` RPM をダウンロードし、その RPM を追加します。

```
install add source [repository]
<platform>-sysadmin-hostos-6.5.1.06-r65108I.CSChu44444.host.arm
<platform>-sysadmin-hostos-6.5.1.06-r65108I.CSChu44444.admin.arm
<platform>-sysadmin-hostos-6.5.1.06-r65108I.CSChu44444.host.x86_64
<platform>-sysadmin-hostos-6.5.1.06-r65108I.CSChu44444.admin.x86_64
```

2. ダウンロードした RPM をアクティブにします。

```
install activate [repository]
<platform>-sysadmin-hostos-6.5.1.06-r65108I.CSChu44444.host.arm
<platform>-sysadmin-hostos-6.5.1.06-r65108I.CSChu44444.admin.arm
<platform>-sysadmin-hostos-6.5.1.06-r65108I.CSChu44444.host.x86_64
<platform>-sysadmin-hostos-6.5.1.06-r65108I.CSChu44444.admin.x86_64
```

3. 設定をコミットします。

```
install commit
```

- シナリオ 2 : ベース RPM をアクティブにして `hostos` RPM を非アクティブにします。バージョン 09 がアクティブであると見なします。

1. ベース RPM をアクティブにします。

```
install activate <platform>-sysadmin-hostos-6.5.1.08I-r65108I.admin.arm
<platform>-sysadmin-hostos-6.5.1.08I-r65108I.host.arm
```



```
<platform>-sysadmin-hostos-6.5.1.08I-r65108I.admin.x86_64  
<platform>-sysadmin-hostos-6.5.1.08I-r65108I.host.x86_64
```

たとえば、RPM がインストール済みの RPM の場合、はベース RPM です。

2. 設定をコミットします。

```
install commit
```

サードパーティ製の RPM のダウングレードは、hostos RPM と同様です。SMU をダウングレードするには、SMU の下位バージョンをアクティブにします。存在する SMU のバージョンが 1 つのみの場合は、その SMU のベース RPM をアクティブにする必要があります。



-
- (注) hostos とサードパーティ製の RPM を非アクティブにすることはできません。異なるバージョンのアクティブバージョンのみがサポートされています。
-

• hostos 以外の RPM のダウングレード

1. RPM を非アクティブにして、以前のバージョンの RPM にダウングレードします。

```
install deactivate <platform>-<rpm-name>
```

2. RPM のアクティブなバージョンを確認します。

```
show install active
```

3. 設定をコミットします。

```
install commit
```




第 8 章

ゴールデンISOを使用したインストールのカスタマイズ

表 6: 機能の履歴 (表)

機能名	リリース情報	説明
バグ修正 RPM のブリッジングの自動インストール	リリース 7.5.1	この機能により、GISO に基づいて、ワンステップかつプロンプトなしで簡単にアップグレードまたはダウングレードできるため、アップグレードまたはダウングレードを実行する前に、バグ修正 RPM のブリッジングを手動でインストールする必要がなくなります。

ゴールデンISO (GISO) は、ユーザーがインストール要件に合わせて作成できるカスタマイズ ISO です。ユーザーはインストール可能なイメージをカスタマイズして、基本的な機能コンポーネントを備えた標準的な基本イメージを含め、要件に基づいて RPM、SMU、および設定ファイルを追加することができます。

インストールが簡単であること、およびシステムをシームレスにインストールまたはアップグレードするためにかかる時間は、クラウド規模のネットワークで重要な役割を果たします。時間がかかる複雑なインストールプロセスは、ネットワークの復元力と拡張性に影響します。GISO によってインストールプロセスが簡素化され、インストールワークフローが自動化され、RPM および SMU の依存関係が自動的に管理されます。

GISO は、github の場所 ([Github](#) の場所) で利用可能なスクリプト `gisobuild.py` を使用して作成します。

Cisco IOS XR リリース 7.5.1 以降、バグ修正 RPM のブリッジングの自動インストール機能を使用して、システムアップグレードまたはダウングレードの前提条件であるバグ修正 RPM のブリッジングをインストールできます。シスコゴールデンISO (GISO) ビルドスクリプト `gisobuild.py` を使用してカスタマイズされた ISO ビルドに、必要なバグ修正 RPM のブリッジ

グを追加する必要があります。GISO には、複数のリリースのバグ修正 RPM のブリッジングを含めることができますが、ターゲットリリースに必要な特定のバグ修正 RPM のブリッジングのみをインストールします。バグ修正 RPM のブリッジングは、次のシナリオで使用できます。

- アップグレードを停止する可能性のあるバグを解決するため、
- 最新バージョンには、以前のバージョンでは満たされていない新しい前提条件の要件があります。

システムを GISO を使用して起動すると、GISO 内の追加の SMU および RPM が自動的にインストールされ、ルータは GISO の XR 設定で事前に設定されます。GISO のダウンロードおよびインストールの詳細については、[ゴールデン ISO のインストール \(92 ページ\)](#) を参照してください。

GISO の機能は次の場合に使用できます。

- IOS XR 32 ビットから IOS XR 64 ビットへの移行
- ルータの初期展開
- ソフトウェア ディザスタ リカバリ
- 1 つの基本バージョンから別のバージョンへのシステム アップグレード
- 追加 SMU を使用した同じ基本バージョンからのシステム アップグレード
- 更新プログラムのインストールと依存パッケージの識別および更新
- [制限事項 \(80 ページ\)](#)
- [ゴールデン ISO を使用したインストールのカスタマイズ \(79 ページ\)](#)
- [ゴールデン ISO ワークフロー \(81 ページ\)](#)
- [ゴールデン ISO の構築 \(82 ページ\)](#)
- [ゴールデン ISO のインストール \(92 ページ\)](#)

制限事項

次に、カスタム ISO に関する既知の問題および制限事項を示します。

- サイズが 1.8 GB を超える GISO イメージはサポートされていません。RSP880-LT-SE/TR の最大イメージサイズは 1.599 GB です。
- 非同期パッケージの GISO (ISO とは異なるリリースのパッケージ) の作成と起動はサポートされていません。
- GISO 作成スクリプト `gisobuild.py` は XR 設定の確認をサポートしていません。
- GISO ビルドの名前を変更し、その名前を変更した GISO ビルドのインストールはサポートされていません。
- IPv6 を介したインストール操作はサポートされていません。

- GISO を使用した IOS XR 32 ビットから 64 ビット OS への移行には、次の制限があります。
 - IOS XR 32 ビットから 64 ビットへの変換スクリプトでは、48 文字を超えるファイル名はサポートされていません。
 - IOS XR 32 ビット OS の最大ファイルサイズの制限は 2 GB です。GISO のサイズがこの制限を超えないようにしてください。
- 移行方法とシステム要件の詳細については、[Cisco ASR 9000 シリーズルータの移行ガイド \[英語\]](#) を参照してください。

ゴールデン ISO を使用したインストールのカスタマイズ

表 7: 機能の履歴 (表)

機能名	リリース情報	説明
バグ修正 RPM のブリッジングの自動インストール	リリース 7.5.1	この機能により、GISO に基づいて、ワンステップかつプロンプトなしで簡単にアップグレードまたはダウングレードできるため、アップグレードまたはダウングレードを実行する前に、バグ修正 RPM のブリッジングを手動でインストールする必要がなくなります。

ゴールデン ISO (GISO) は、ユーザーがインストール要件に合わせて作成できるカスタマイズ ISO です。ユーザーはインストール可能なイメージをカスタマイズして、基本的な機能コンポーネントを備えた標準的な基本イメージを含め、要件に基づいて RPM、SMU、および設定ファイルを追加することができます。

インストールが簡単であること、およびシステムをシームレスにインストールまたはアップグレードするためにかかる時間は、クラウド規模のネットワークで重要な役割を果たします。時間がかかる複雑なインストールプロセスは、ネットワークの復元力と拡張性に影響します。GISO によってインストールプロセスが簡素化され、インストールワークフローが自動化され、RPM および SMU の依存関係が自動的に管理されます。

GISO は、github の場所 ([Github](#) の場所) で利用可能なスクリプト `gisobuild.py` を使用して作成します。

Cisco IOS XR リリース 7.5.1 以降、バグ修正 RPM のブリッジングの自動インストール機能を使用して、システムアップグレードまたはダウングレードの前提条件であるバグ修正 RPM のブリッジングをインストールできます。シスコゴールデン ISO (GISO) ビルドスクリプト `gisobuild.py` を使用してカスタマイズされた ISO ビルドに、必要なバグ修正 RPM のブリッジ

グを追加する必要があります。GISOには、複数のリリースのバグ修正RPMのブリッジングを含めることができますが、ターゲットリリースに必要な特定のバグ修正RPMのブリッジングのみをインストールします。バグ修正RPMのブリッジングは、次のシナリオで使用できます。

- アップグレードを停止する可能性のあるバグを解決するため、
- 最新バージョンには、以前のバージョンでは満たされていない新しい前提条件の要件があります。

システムを GISO を使用して起動すると、GISO 内の追加の SMU および RPM が自動的にインストールされ、ルータは GISO の XR 設定で事前に設定されます。GISO のダウンロードおよびインストールの詳細については、[ゴールデン ISO のインストール \(92 ページ\)](#) を参照してください。

GISO の機能は次の場合に使用できます。

- IOS XR 32 ビットから IOS XR 64 ビットへの移行
- ルータの初期展開
- ソフトウェア ディザスタ リカバリ
- 1 つの基本バージョンから別のバージョンへのシステム アップグレード
- 追加 SMU を使用した同じ基本バージョンからのシステム アップグレード
- 更新プログラムのインストールと依存パッケージの識別および更新

制限事項

次に、カスタム ISO に関する既知の問題および制限事項を示します。

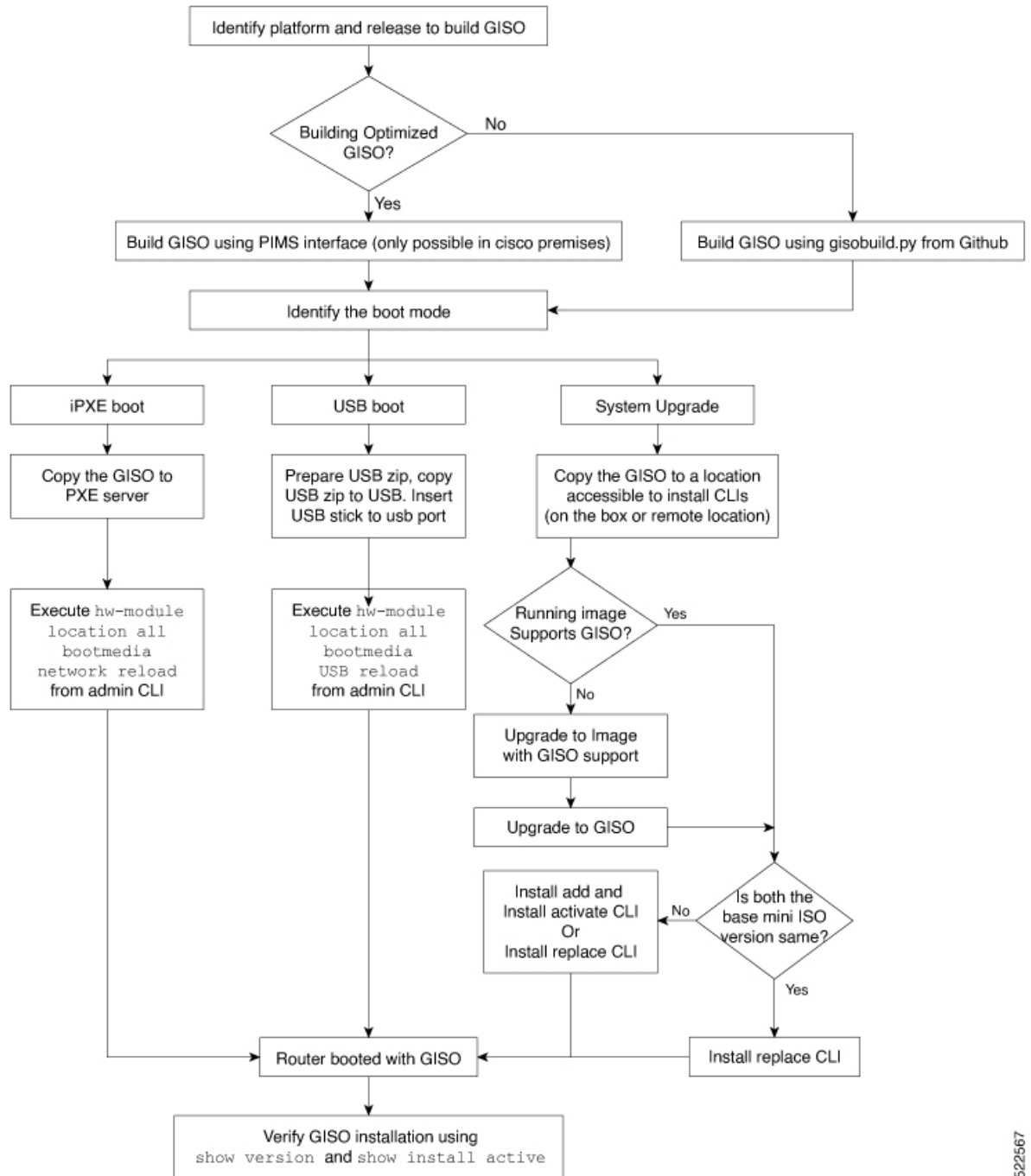
- サイズが 1.8 GB を超える GISO イメージはサポートされていません。RSP880-LT-SE/TR の最大イメージサイズは 1.599 GB です。
- 非同期パッケージの GISO (ISO とは異なるリリースのパッケージ) の作成と起動はサポートされていません。
- GISO 作成スクリプト `gisobuild.py` は XR 設定の確認をサポートしていません。
- GISO ビルドの名前を変更し、その名前を変更した GISO ビルドのインストールはサポートされていません。
- IPv6 を介したインストール操作はサポートされていません。
- GISO を使用した IOS XR 32 ビットから 64 ビット OS への移行には、次の制限があります。
 - IOS XR 32 ビットから 64 ビットへの変換スクリプトでは、48 文字を超えるファイル名はサポートされていません。
 - IOS XR 32 ビット OS の最大ファイルサイズの制限は 2 GB です。GISO のサイズがこの制限を超えないようにしてください。

移行方法とシステム要件の詳細については、[Cisco ASR 9000 シリーズルータの移行ガイド \[英語\]](#) を参照してください。

ゴールデン ISO ワークフロー

次の図は、ゴールデン ISO を構築してインストールするためのワークフローを示しています。

図 6: ゴールデン ISO ワークフロー



522567

ゴールデン ISO の構築

カスタマイズした ISO は、[Github](#) の場所で利用可能なシスコゴールデン ISO (GISO) 作成スクリプト `gisobuild.py` を使用して構築します。

GISO 作成スクリプトは、自動依存関係管理をサポートし、次の機能を提供します。

- パッケージリポジトリ内に存在するすべてのパッケージの RPM データベースを構築します。
- リポジトリをスキャンし、入力 ISO と一致する関連の Cisco RPM を選択します。
- mini-x.iso 内にすでに存在するサードパーティ製の基本パッケージの SMU ではないサードパーティの RPM をスキップおよび削除します。
- 同じリリースで異なるバージョンの基本 RPM が複数ある場合、エラーを表示し、作成プロセスを終了します。
- すべての RPM の互換性チェックと依存関係チェックを実行します。たとえば、子 RPM asr9k-mpls-te-rsvp は親 RPM asr9k-mpls に依存します。子 RPM のみが含まれる場合、ゴールデン ISO の作成は失敗します。

スクリプトを使用したゴールデン ISO の構築

表 8: 機能の履歴 (表)

機能名	リリース情報	説明
強化されたゴールデン ISO ビルドツール	リリース 7.5.1	この拡張機能により、gisobuild.py ツールを使用して、カスタマイズされたインストール要件に合わせて Cisco IOS XR ソフトウェアコマンド、YAML ベースのテンプレートファイル、または Docker 機能を使用して GISO イメージを柔軟に構築できます。GISO を作成する場合は、基本イメージとオプションの RPM を使用してルータを自動的にプロビジョニングすることに加えて、ゼロタッチプロビジョニング (ZTP) 初期化ファイル、スクリプト初期化ファイル、Cisco IOS XR 構成ファイル、および SMU の指定もできます。

GISO を作成するには、スクリプトに次の入力パラメータを指定します。

- 基本 mini-x.iso (必須)
- XR コンフィギュレーション ファイル (任意)

- ホスト、XR、およびシステム管理用の 1 つまたは複数のシスコ固有の SMU（任意）
- ホスト、XR、およびシステム管理用の 1 つまたは複数のサードパーティ製 SMU（任意）
- ゴールデン ISO のラベル（任意）
- オプションの RPM
- ZTP 初期化 `ztp.ini` ファイル（任意）
- スクリプト初期化 `script.ini` ファイル（任意）

GISO スクリプトは XR 設定の検証をサポートしていません。



- (注) `k9sec` RPM を GISO に適切に追加するには、`chmod` コマンドを使用してファイルの権限を `644` に変更します。

```
chmod 644 [k9 sec rpm]
```

Cisco IOS XR リリース 7.5.1 では、`gisobuild.py` GISO ビルドツールの拡張機能が導入されています。`ztp.ini` ZTP 初期化ファイルと `script.ini` スクリプト初期化ファイルも追加できます。ZTP 設定は、現在のソフトウェアバージョンが GISO イメージを使用するバージョンに置き換えられたとき、またはロールバックされたときにルータに適用され、ルータを自動的にプロビジョニングするために ZTP が実行されるたびに使用されます。このビルドツールでは複数のリポジトリがサポートされています。CLI コマンド、Docker、または YAML ファイルを使用して GISO を構築できます。



- (注)
- Cisco ASR 9000 シリーズルータの IOS XR 32 ビットソフトウェアから IOS XR 64 ビットソフトウェアに移行する場合は、`[migration]` オプションを `[true]` に設定します。
 - 最初の GISO を作成後に同じビルドディレクトリを使用する場合は、`[clean]` オプションを `[true]` に設定します。後続のすべての GISO ビルドでオプションを `[true]` に設定していることを確認します。
 - Docker を使用して GISO を構築する場合は、`[docker]` オプションを `[true]` に設定します。
 - GISO 構築時のエラーを回避するために、YAML ファイルの形式と構文が変更されていないことを確認します。たとえば、`:` 記号が欠落している場合、またはサポートされていない記号がテンプレートで使用されている場合、GISO の構築時にエラーが表示されます。

`gisobuild.py` ツールはネイティブに実行できます。また、Docker サービスが有効になっている、公開された Docker イメージをプルできるシステムで実行できます。追加の権限は不要なため、Docker を使用してイメージを構築することを推奨します。



- (注) [full-iso] オプションは、Cisco IOS XRv 9000 ルータに固有の完全な ISO イメージ `xrv9k-full-x-7.5.1.iso` を作成するために使用されます。Cisco IOS XR リリース 7.8.1 以降では、GISO の構築には完全な ISO イメージを使用できません。

GISO を作成するには、次の手順を実行します。

始める前に

- GISO が構築されているシステムは、次の要件を満たしている必要があります。
 - システムには Python バージョン 3.6 以降が必要です。
 - システムには、最低 12 GB の空きディスク容量が必要です。
 - システムに Linux ユーティリティ `mount`、`rm`、`cp`、`umount`、`zcat`、`chroot`、`mkisofs` があることを確認します。これらのユーティリティはスクリプトによって使用されます。これらすべての Linux コマンドを実行する権限があることを確認します。ただし、`Docker` を使用している場合は、これらのユーティリティは不要です。
 - システムのカーネルバージョンは、Cisco ISO のカーネルバージョンより後の 3.16 以降である必要があります。
 - Linux カーネルでサポートされている `libyaml rpm` が、ツールで `yaml` を正常に実行できることを確認します。
 - ユーザーは `rpm` リポジトリのセキュリティ `rpm` (`k9sec-rpm`) に対する適切な権限を持っている必要があります。それ以外の場合は、ゴールデン ISO の作成でセキュリティ `rpm` が無視されます。
- `gisobuild.py` スクリプトが実行されるシステムには、ルートログイン情報が必要ですが、`Docker` コンテナ内でイメージを構築する場合は必須ではありません。
- `gisobuild.py` スクリプトを使用する前に `git pull` 操作を実行して、Python バージョン用のスクリプトの最新バージョンを取得することを推奨します。

ステップ 1 スクリプト `gisobuild.py` を [GitHub](#) リポジトリから、GISO を構築するオフラインシステムまたは外部サーバーにコピーします。このシステムが上記の「始める前に」セクションに記載された前提条件を満たしていることを確認します。

ステップ 2 スクリプト `gisobuild.py` を実行し、ルータからゴールデン ISO を作成するためのパラメータを指定します。すべての RPM と SMU が同じディレクトリ内または 1 つのリポジトリに存在することを確認します。ゴールデン ISO の作成に使用できる RPM と SMU の数は 64 です。

```
usage: gisobuild.py [-h] [--iso ISO] [--repo REPO [REPO ...]]
                  [--bridging-fixes BRIDGE_FIXES [BRIDGE_FIXES ...]]
                  [--xrconfig XRCONFIG] [--ztp-ini ZTP_INI] [--label LABEL]
                  [--out-directory OUT_DIRECTORY] [--yamlfile CLI_YAML] [--clean]
                  [--pkglist PKGLIST [PKGLIST ...]] [--script SCRIPT] [--docker]
                  [--x86-only] [--migration]
```

```

[--remove-packages REMOVE_PACKAGES [REMOVE_PACKAGES ...]]
[--skip-usb-image] [--copy-dir COPY_DIRECTORY]
[--clear-bridging-fixes] [--verbose-dep-check] [--debug]
[--version]

```

Utility to build Golden ISO for IOS-XR.

optional arguments:

```

-h, --help          show this help message and exit
--iso ISO           Path to Mini.iso/Full.iso file
--repo REPO [REPO ...]
                    Path to RPM repository. For LNT, user can specify .rpm, .tgz,
                    .tar filenames, or directories. RPMs are only used if already
                    included in the ISO, or specified by the user via the
                    --pkglist option.
--bridging-fixes BRIDGE_FIXES [BRIDGE_FIXES ...]
                    Bridging rpms to package. For EXR, takes from-release or rpm
                    names; for LNT, the user can specify the same file types as for
                    the --repo option.
--xrconfig XRCONFIG Path to XR config file
--ztp-ini ZTP_INI   Path to user ztp ini file
--label LABEL, -l LABEL
                    Golden ISO Label
--out-directory OUT_DIRECTORY
                    Output Directory
--yamlfile CLI_YAML Cli arguments via yaml
--clean            Delete output dir before proceeding
--pkglist PKGLIST [PKGLIST ...]
                    Packages to be added to the output GISO. For eXR: optional rpm
                    or smu to package. For LNT: either full package filenames or
                    package names for user installable packages can be specified.
                    Full package filenames can be specified to choose a particular
                    version of a package, the rest of the block that the package is
                    in will be included as well. Package names can be specified to
                    include optional packages in the output GISO.
--docker, --use-container
                    Build GISO in container environment. Pulls and run pre-built
                    container image to build GISO.
--version          Print version of this script and exit

```

EXR only build options:

```

--script SCRIPT    Path to user executable script executed as part of bootup post
                    activate.
--x86-only         Use only x86_64 rpms even if other architectures are
                    applicable.
--migration        To build Migration tar only for ASR9k

```

LNT only build options:

```

--remove-packages REMOVE_PACKAGES [REMOVE_PACKAGES ...]
                    Remove RPMs, specified in a comma separated list. These are
                    matched against user installable package names, and must be the
                    whole package name, e.g: xr-bgp
--skip-usb-image   Do not build the USB image
--copy-dir COPY_DIRECTORY
                    Copy built artefacts to specified directory if provided. The
                    specified directory must already exist, be writable by the
                    builder and must not contain a previously built artefact with
                    the same name.
--clear-bridging-fixes
                    Remove all bridging bugfixes from the input ISO
--verbose-dep-check
                    Verbose output for the dependency check.
--debug            Output debug logs to console

```

例

例 : Docker ベースの GISO イメージの構築

この例では、GISO イメージは Docker を使用して構築されます。

```
[root@xr src]# ./gisobuild.py --docker --iso /auto/ncs5500giso/ncs5500-mini-x-7.5.1.iso

--repo /auto/ncs5500giso --pkglist ncs5500-bgp-2.0.0.0-r751.x86_64.rpm
ncs5500-eigrp-1.0.0.0-r751.x86_64.rpm ncs5500-isis-2.1.0.0-r751.x86_64.rpm
ncs5500-k9sec-3.1.0.0-r751.x86_64.rpm
ncs5500-li-1.0.0.0-r751.x86_64.rpm ncs5500-mcast-3.0.0.0-r751.x86_64.rpm
ncs5500-mgbl-3.0.0.0-r751.x86_64.rpm
ncs5500-mppls-2.1.0.0-r751.x86_64.rpm ncs5500-mppls-te-rsvp-3.1.0.0-r751.x86_64.rpm
ncs5500-ospf-2.0.0.0-r751.x86_64.rpm
ncs5500-parser-2.0.0.0-r751.x86_64.rpm --label dockerbasedgiso

Local System requirements check [PASS]
Pulling gisobuild image from hub. Please wait...
\
Done...
System requirements check [PASS]

Platform: ncs5500 Version: 7.5.1

Scanning repository [/auto/ncs5500giso]...

Building RPM Database...

Total 11 RPM(s) present in the repository path provided in CLI
[ 1] ncs5500-mppls-2.1.0.0-r751.x86_64.rpm
[ 2] ncs5500-mgbl-3.0.0.0-r751.x86_64.rpm
[ 3] ncs5500-bgp-2.0.0.0-r751.x86_64.rpm
[ 4] ncs5500-parser-2.0.0.0-r751.x86_64.rpm
[ 5] ncs5500-isis-2.1.0.0-r751.x86_64.rpm
[ 6] ncs5500-mcast-3.0.0.0-r751.x86_64.rpm
[ 7] ncs5500-mppls-te-rsvp-3.1.0.0-r751.x86_64.rpm
[ 8] ncs5500-ospf-2.0.0.0-r751.x86_64.rpm
[ 9] ncs5500-li-1.0.0.0-r751.x86_64.rpm
[10] ncs5500-eigrp-1.0.0.0-r751.x86_64.rpm
[11] ncs5500-k9sec-3.1.0.0-r751.x86_64.rpm

Following XR x86_64 rpm(s) will be used for building Golden ISO:

(+) ncs5500-ospf-2.0.0.0-r751.x86_64.rpm
(+) ncs5500-bgp-2.0.0.0-r751.x86_64.rpm
(+) ncs5500-parser-2.0.0.0-r751.x86_64.rpm
(+) ncs5500-mcast-3.0.0.0-r751.x86_64.rpm
(+) ncs5500-li-1.0.0.0-r751.x86_64.rpm
(+) ncs5500-eigrp-1.0.0.0-r751.x86_64.rpm
(+) ncs5500-mgbl-3.0.0.0-r751.x86_64.rpm
(+) ncs5500-mppls-2.1.0.0-r751.x86_64.rpm
(+) ncs5500-mppls-te-rsvp-3.1.0.0-r751.x86_64.rpm
(+) ncs5500-isis-2.1.0.0-r751.x86_64.rpm
(+) ncs5500-k9sec-3.1.0.0-r751.x86_64.rpm

...RPM signature check [PASS]

Skipping following rpms from repository since they are already present in base ISO:

(-) ncs5500-parser-2.0.0.0-r751.x86_64.rpm
(-) ncs5500-bgp-2.0.0.0-r751.x86_64.rpm
```

```

...RPM compatibility check [PASS]

Building Golden ISO...
Summary .....

XR rpms:
ncs5500-mcast-3.0.0.0-r751.x86_64.rpm
ncs5500-mgbl-3.0.0.0-r751.x86_64.rpm
ncs5500-isis-2.1.0.0-r751.x86_64.rpm
ncs5500-mppls-te-rsvp-3.1.0.0-r751.x86_64.rpm
ncs5500-eigrp-1.0.0.0-r751.x86_64.rpm
ncs5500-mppls-2.1.0.0-r751.x86_64.rpm
ncs5500-ospf-2.0.0.0-r751.x86_64.rpm
ncs5500-li-1.0.0.0-r751.x86_64.rpm
ncs5500-k9sec-3.1.0.0-r751.x86_64.rpm

...Golden ISO creation SUCCESS.

Golden ISO Image Location: /var/tmp/giso/gisobuild-toolkit-master/src/output_gisobuild/
ncs5500-golden-x-7.5.1-dockerbasedgiso.iso

[root@xr src]# ./gisobuild.py --docker --iso /auto/asr9kgiso/asr9k-mini-x-7.5.1.iso
--repo /auto/asr9kgiso --pkglst asr9k-7.5.1.CSCsb88888 asr9k-bgp-2.0.0.0-r751.x86_64.rpm
asr9k-eigrp-1.0.0.0-r751.x86_64.rpm asr9k-isis-2.1.0.0-r751.x86_64.rpm
asr9k-k9sec-3.1.0.0-r751.x86_64.rpm
asr9k-li-1.0.0.0-r751.x86_64.rpm asr9k-mcast-3.0.0.0-r751.x86_64.rpm
asr9k-mgbl-3.0.0.0-r751.x86_64.rpm
asr9k-mppls-2.1.0.0-r751.x86_64.rpm asr9k-mppls-te-rsvp-3.1.0.0-r751.x86_64.rpm
asr9k-ospf-2.0.0.0-r751.x86_64.rpm
asr9k-parser-2.0.0.0-r751.x86_64.rpm --label dockerbasedgiso

Local System requirements check [PASS]
Pulling gisobuild image from hub. Please wait...
\
Done...
System requirements check [PASS]

Platform: asr9000 Version: 7.5.1

Scanning repository [/auto/asr9000giso]...

Building RPM Database...

Total 11 RPM(s) present in the repository path provided in CLI
[ 1] asr9k-mppls-2.1.0.0-r751.x86_64.rpm
[ 2] asr9k-mgbl-3.0.0.0-r751.x86_64.rpm
[ 3] asr9k-bgp-2.0.0.0-r751.x86_64.rpm
[ 4] asr9k-parser-2.0.0.0-r751.x86_64.rpm
[ 5] asr9k-isis-2.1.0.0-r751.x86_64.rpm
[ 6] asr9k-mcast-3.0.0.0-r751.x86_64.rpm
[ 7] asr9k-mppls-te-rsvp-3.1.0.0-r751.x86_64.rpm
[ 8] asr9k-ospf-2.0.0.0-r751.x86_64.rpm
[ 9] asr9k-li-1.0.0.0-r751.x86_64.rpm
[10] asr9k-eigrp-1.0.0.0-r751.x86_64.rpm
[11] asr9k-k9sec-3.1.0.0-r751.x86_64.rpm
Following XR x86_64 rpm(s) will be used for building Golden ISO:

(+) asr9k-ospf-2.0.0.0-r751.x86_64.rpm
(+) asr9k-bgp-2.0.0.0-r751.x86_64.rpm
(+) asr9k-parser-2.0.0.0-r751.x86_64.rpm
(+) asr9k-mcast-3.0.0.0-r751.x86_64.rpm
(+) asr9k-li-1.0.0.0-r751.x86_64.rpm

```

```

(+) asr9k-eigrp-1.0.0.0-r751.x86_64.rpm
(+) asr9k-mgbl-3.0.0.0-r751.x86_64.rpm
(+) asr9k-mpls-2.1.0.0-r75114I.x86_64.rpm
(+) asr9k-mpls-te-rsvp-3.1.0.0-r751.x86_64.rpm
(+) asr9k-isis-2.1.0.0-r751.x86_64.rpm
(+) asr9k-k9sec-3.1.0.0-r751.x86_64.rpm

...RPM signature check [PASS]

Skipping following rpms from repository since they are already present in base ISO:

(-) asr9k-parser-2.0.0.0-r751.x86_64.rpm
(-) asr9k-bgp-2.0.0.0-r751.x86_64.rpm

...RPM compatibility check [PASS]

Building Golden ISO...
Summary .....

XR rpms:
asr9k-mcast-3.0.0.0-r751.x86_64.rpm
asr9k-mgbl-3.0.0.0-r751.x86_64.rpm
asr9k-isis-2.1.0.0-r751.x86_64.rpm
asr9k-mpls-te-rsvp-3.1.0.0-r751.x86_64.rpm
asr9k-eigrp-1.0.0.0-r751.x86_64.rpm
asr9k-mpls-2.1.0.0-r751.x86_64.rpm
asr9k-ospf-2.0.0.0-r751.x86_64.rpm
asr9k-li-1.0.0.0-r751.x86_64.rpm
asr9k-k9sec-3.1.0.0-r751.x86_64.rpm

...Golden ISO creation SUCCESS.

Golden ISO Image Location: /var/tmp/giso/gisobuild-toolkit-master/src/output_gisobuild/
                           asr9k-golden-x-7.5.1-dockerbasedgiso.iso

```

GISO ファイルが正常に作成されたことを確認します。

```

[root@xr src]# ls
exrmod  gisobuild.py  lntmod  output_gisobuild  utils

[root@xr src]# cd output_gisobuild/
[root@xr output_gisobuild]# ls
img_built_name.txt  logs  ncs5500asr9k-golden-x-7.5.1-dockerbasedgiso.iso
rpms_packaged_in_giso.txt

```

例：YAML ベースの GISO イメージの構築

YAML は、パッケージリストを提供し、ビルドオプションを管理するためのテンプレートとして機能するマークアップファイルです。

次に、サンプルの YAML テンプレートの例を示します。

```

# Options below correspond to the tool input options.
# --iso ISO                Path to Mini.iso/golden.iso file
# --repo REPO [REPO ...]
#
#                          Path to list of RPM repositories. RPMs are only used if already
#                          included in the ISO, or specified by the user via the --pkglist
#                          option.
# --pkglist PKGLIST [PKGLIST ...]
#                          Optional list of rpm or smu to add to the ISO.
# --remove-packages REMOVE_PACKAGES [REMOVE_PACKAGES ...]
#                          Remove named RPMs, specified in a space separated list. Valid
#                          build
#                          option for LNT only. eXR builds simply ignores this option.

```

```

# --bridging-fixes BRIDGE_FIXES [BRIDGE_FIXES ...]
#                               Bridging rpms to package. Takes from-release (supported for eXR)
#                               or rpm names.
# --xrconfig XRCONFIG           Path to XR config file
# --ztp-ini ZTP_INI             Path to user ztp ini file
# --script SCRIPT               Path to user executable script executed as part of
#                               bootup post activate. Valid build option for eXR only.
#                               LNT builds simply ignores.
# --label LABEL                 Golden ISO Label
# --out-directory OUT_DIRECTORY
#                               Output Directory. Built GISO and logs will be available post
gisobuild.
# --copy-directory COPY_DIRECTORY
#                               Copy built artefacts to specified directory if provided. Valid
build
#                               option for LNT only. eXR build ignores this option.
# --yamlfile CLI_YAML           Cli arguments via yaml.
# --clean                       Delete output dir before proceeding.
# --migration                   To build Migration tar only for ASR9k. Valid build option for
eXR only.
#                               LNT builds simply ignore this option.
# --docker                      Load and run pre-built docker image. Valid build option for eXR
only.
#                               LNT builds simply ignore this option.
# --x86-only                    Use only x86_64 rpms even if other architectures are applicable.
Valid build
#                               option for eXR only. LNT builds simply ignore this option.
# --version                     Print version of this script and exit

packages:
  iso: <path-to-iso>
  repo:
    - <path-to-repo1>
    - <path-to-repo2>
  pkglist:
    - <pkg1>
    - <pkg2>
  bridge-fixes:
    upgrade-from-release:
      - <dotted-release-1>
      - <dotted-release-2>
    rpms:
      - <pkg1>
      - <pkg2>
  remove_packages:
    - <pkg1>
    - <pkg2>

user-content:
  script: <path-to-script-sh>
  xrconfig: <path-to-router.cfg>
  ztp-ini: <path-to-ztp.ini>

output:
  label: <giso-label>
  out-directory: <path-to-output-directory>
  clean: <true/false>

options:
  docker: <true/false>
  migration: <true/false>
  x86-only: <true/false>

```

この例では、必要なファイルを指定して YAML ファイルを設定します。


```

packages:
  iso: /auto/751_repo/ncs5500-mini-x-7.5.1.iso
  repo:
    - /auto/751_repo/
  pkglist:
    - ncs5500-bgp-2.0.0.0-r751.x86_64.rpm
    - ncs5500-eigrp-1.0.0.0-r751.x86_64.rpm
    - ncs5500-isis-2.1.0.0-r751.x86_64.rpm
    - ncs5500-li-1.0.0.0-r751.x86_64.rpm
    - ncs5500-mcast-3.0.0.0-r751.x86_64.rpm
    - ncs5500-mgbl-3.0.0.0-r751.x86_64.rpm
    - ncs5500-mpls-2.1.0.0-r751.x86_64.rpm
    - ncs5500-mpls-te-rsvp-3.1.0.0-r751.x86_64.rpm
    - ncs5500-ospf-2.0.0.0-r751.x86_64.rpm
    - ncs5500-parser-2.0.0.0-r751.x86_64.rpm
    - ncs5500-k9sec-3.1.0.0-r751.x86_64.rpm
    - ncs5500-mcast-3.0.0.1-r751.CSCxr33333.x86_64.rpm
    - ncs5500-os-5.0.0.1-r751.CSCxr11111.x86_64.rpm
    - ncs5500-sysadmin-hostos-7.5.1-r751.CSCho99999.admin.x86_64.rpm
    - ncs5500-sysadmin-hostos-7.5.1-r751.CSCho99999.host.x86_64.rpm
    - ncs5500-sysadmin-topo-7.5.1-r751.CSCcv55555.x86_64.rpm
    - ncs5500-sysadmin-system-7.5.1-r751.CSCcv44444.x86_64.rpm
    - openssh-scp-6.6p1.pl-r0.5.0.r751.CSCTp11111.xr.x86_64.rpm
    - cisco-klm-zermatt-0.1.pl-r0.0.r751.CSCTp11111.xr.x86_64.rpm

  remove_rpms: []

user-content:
  script: script.sh
  xrconfig: /auto/751_repo/gisoxrconfig.cfg
  ztp-ini: /auto/751_repo/ztp.ini

output:
  label: 751_yaml_install
  out-directory: /auto/751_repo/
  clean: true

options:
  docker: false
  full-iso: false
  migration: false
  x86-only: false

packages:
  iso: /auto/751_repo/asr9k-mini-x-7.5.1.iso
  repo:
    - /auto/751_repo/
  pkglist:
    - asr9k-bgp-2.0.0.0-r751.x86_64.rpm
    - asr9k-eigrp-1.0.0.0-r751.x86_64.rpm
    - asr9k-isis-2.1.0.0-r751.x86_64.rpm
    - asr9k-li-1.0.0.0-r751.x86_64.rpm
    - asr9k-mcast-3.0.0.0-r751.x86_64.rpm
    - asr9k-mgbl-3.0.0.0-r751.x86_64.rpm
    - asr9k-mpls-2.1.0.0-r751.x86_64.rpm
    - asr9k-mpls-te-rsvp-3.1.0.0-r751.x86_64.rpm
    - asr9k-ospf-2.0.0.0-r751.x86_64.rpm
    - asr9k-parser-2.0.0.0-r751.x86_64.rpm
    - asr9k-k9sec-3.1.0.0-r751.x86_64.rpm
    - asr9k-mcast-3.0.0.1-r751.CSCxr33333.x86_64.rpm
    - asr9k-os-5.0.0.1-r751.CSCxr11111.x86_64.rpm
    - asr9k-sysadmin-hostos-7.5.1-r751.CSCho99999.admin.x86_64.rpm
    - asr9k-sysadmin-hostos-7.5.1-r751.CSCho99999.host.x86_64.rpm
    - asr9k-sysadmin-topo-7.5.1-r751.CSCcv55555.x86_64.rpm
    - asr9k-sysadmin-system-7.5.1-r751.CSCcv44444.x86_64.rpm
  
```

```

- openssh-scp-6.6p1.p1-r0.5.0.r751.CSCTp11111.xr.x86_64.rpm
- cisco-klm-zermatt-0.1.p1-r0.0.r751.CSCTp11111.xr.x86_64.rpm

remove_rpms: []

user-content:
  script: script.sh
  xrconfig: /auto/751_repo/gisoxrconfig.cfg
  ztp-ini: /auto/751_repo/ztp.ini

output:
  label: 751_yaml_install
  out-directory: /auto/751_repo/
  clean: true

options:
  docker: false
  full-iso: false
  migration: false
  x86-only: false

```

CLI を使用してパッケージとパラメータのリストを指定しない場合は、YAML ファイルテンプレートを 사용할 ことができます。

```
[directory-path]$ ./src/gisobuild.py --yamlfile <input-yaml-cfg>
```

YAML 構成ファイルの入力をオーバーライドするには、対応する CLI オプションを使用します。

```
[directory-path]$ ./src/gisobuild.py --yamlfile <input-yaml-cfg> --label <new-label>
```

この新しいラベルで、YAML ファイルで指定されたラベルがオーバーライドされます。

ホストマシンでパッケージの依存関係が満たされていないが、Docker イメージのプルと実行が許可されている場合は、YAML ファイルの [docker] オプションを [true] に設定して、次のコマンドを実行します。

```
[directory-path]$ ./src/gisobuild.py --yamlfile <input-yaml-cfg>
```

ここで、input-yaml-cfg の [docker] オプションは [true] に設定されています。

次のタスク

ルータに GISO イメージをインストールします。

ゴールデン ISO のインストール

ゴールデン ISO (GISO) は、次のアクションを自動的に実行します。

- ホストおよびシステム管理 RPM をインストールします。
- RP でリポジトリと TFTP ブートをパーティションに分割します。
- システム管理モードおよび XR モードでソフトウェア プロファイルを作成します。

- XR RPM をインストールします。 **show install active** コマンドを使用して RPM のリストを表示します。
- XR 設定を適用します。 XR モードで **show running-config** コマンドを使用して確認します。

ステップ 1 次のいずれかのオプションを使用して、ルータに GISO イメージをダウンロードします。

- **PXE ブート** : ルータが起動すると、ブートモードが識別されます。 PXE をブートモードとして検出すると、利用可能なすべてのイーサネットインターフェイスが起動し、各インターフェイスで DHCPClient が実行されます。 DHCPClient スクリプトは HTTP または TFTP プロトコルを解析し、GISO がボックスにダウンロードされます。

PXE ブートモードを使用してルータを起動すると、既存の設定が削除されます。 パーマネントライセンス予約 (PLR) などのスマートライセンス設定を回復するには、ルータの起動後にこれらの設定を有効にします。

```
Router#configure
Router(config)#license smart reservation
Router(config)#commit
```

次に、PXE ブートを使用した GISO のインストールのログの例を示します。

```
...
Fri Dec 2 19:18:03 UTC 2016: ---Starting to prepare host logical volume---
...
Fri Dec 02 19:18:14 UTC 2016: Skipping tp base rpm(openssh-scp-6.6p1-r0.0.host.x86_64.rpm) from
installation
Fri Dec 02 19:18:14 UTC 2016: Skipping tp base rpm(kernel-modules-3.14-r0.1.host.x86_64.rpm)
from installation
Fri Dec 02 19:18:15 UTC 2016: Installing asr9k-sysadmin-hostos-6.1.3-r613.CSChu77777.host.x86_64
[SUCCESS]
...
Fri Dec 2 19:18:23 UTC 2016: ---Starting to prepare calvados logical volume---
...
Fri Dec 02 19:18:48 UTC 2016: Skipping tp base rpm(openssh-scp-6.6p1-r0.0.admin.x86_64.rpm) from
installation
Fri Dec 02 19:18:48 UTC 2016: Skipping tp base rpm(kernel-modules-3.14-r0.1.admin.x86_64.rpm)
from installation
Fri Dec 02 19:18:49 UTC 2016: Installing asr9k-sysadmin-system-6.1.3-r613.CSCcv44444.x86_64
[SUCCESS]
Fri Dec 02 19:18:50 UTC 2016: Installing asr9k-sysadmin-shared-6.1.3-r613.CSCcv33333.x86_64
[SUCCESS]
Fri Dec 02 19:18:51 UTC 2016: Installing asr9k-sysadmin-hostos-6.1.3-r613.CSChu77777.admin.x86_64
[SUCCESS]
...
Fri Dec 2 19:19:07 UTC 2016: ---Starting to prepare repository---
Fri Dec 2 19:19:11 UTC 2016: File system creation on /test took 3 seconds
Fri Dec 2 19:19:11 UTC 2016: Copying /iso/host.iso to repository /iso directory
Fri Dec 2 19:19:11 UTC 2016: Copy Host rpms to repository
Fri Dec 2 19:19:13 UTC 2016: Copying /iso/asr9k-sysadmin.iso to repository /iso directory
Fri Dec 2 19:19:13 UTC 2016: Copy Sysadmin rpms to repository
Fri Dec 2 19:19:16 UTC 2016: Copy HostOs rpms to repository
Fri Dec 2 19:19:16 UTC 2016: Copy XR rpms to repository
```

```
Fri Dec 2 19:19:16 UTC 2016: Copy giso_info.txt to repository
Fri Dec 2 19:19:17 UTC 2016: Copying /iso/asr9k-xr.iso to repository /iso directory
Fri Dec 2 19:19:21 UTC 2016: Copying all ISOs to repository took 10 seconds
```

...

- **USB ブートまたはディスク ブート** : ブート中に USB モードが検出され、GISO が識別されると、追加の RPM および XR 設定ファイルが抽出されてインストールされます。
- **システムアップグレード** : システムアップグレード時には、**install add**、**install activate**、または **install replace** コマンドを使用して GISO をインストールできます。

重要 ルータ上の現在のバージョンとパッケージを GISO のバージョンと置き換えるには、コマンドとフォーマットの変更に注意してください。

- Cisco IOS XR リリース 6.3.3、6.4.x、および 6.5.1 よりも前のバージョンでは、次の **install update** コマンドを使用します。

```
install update source <source path> <Golden-ISO-name> replace
```

- Cisco IOS XR リリース 6.5.2 以降では、**install replace** コマンドを使用します。

```
install replace <absolute-path-of-Golden-ISO>
```

(注) ブート可能な外部 USB ディスクを作成するには、次の手順を実行します。

- USB ブートディスクに 8GB の最小ストレージがあること、および Linux マシンでブート可能なディスクを作成するための root/admin 権限または適切な権限があることを確認します。

1. Linux マシンで usb-install スクリプトをコピーして実行し、ブート可能な外部 USB を作成します。

```
Router#admin

sysadmin-vm:0_RSP0#run chvrf 0 ssh rp0_admin
[sysadmin-vm:0_RSP0:~]$ssh my_host
[host:~]$cd /misc/disk1/
[host:~]$. /usb-install-712-or-latest.sh asr9k-goldenk9-x64-7.0.2-dr.iso /dev/sdc
EFI

Preparing USB stick for EFI
parted gpt: Failed to create partition - continuing ...
Create filesystem on /dev/sdc1
Mounting source iso at //misc/disk1/cdtmp.CnuKnA
Mounting destination /dev/sdc1 at //misc/disk1/usbdev.SSBb4R
Copying image to USB stick
Initrd path is //misc/disk1/cdtmp.CnuKnA/boot/initrd.img
Getting boot
3749342 blocks
Copying boot
Copying initrd.img
Copying signature.initrd.img
Copying certs
Creating grub files
Copying /misc/disk1/asr9k-goldenk9-x64-7.0.2-dr.iso in USB Stick
USB stick set up for EFI boot!
```

2. RSP/RP をリセットし、ブート可能な USB を RSP/RP の前面パネルに接続します。USB は ROMMON で検出されます。システムが ROMMON で起動している場合、RSP/RP がリセットされるまで、前面パネルの外部 USB を追加しても USB は検出されません。

次に、システムをアップグレードするオプションを示します。

- **非 GISO (GISO をサポートしていないイメージ) から GISO イメージへのシステムアップグレード** : システムが GISO をサポートしていないイメージを使用してバージョン 1 を実行している場合、システムは GISO をサポートするイメージのバージョン 2 に直接アップグレードすることはできません。その代わりに、バージョン 1 をバージョン 2 ミニ ISO にアップグレードし、次にバージョン 2 GISO にアップグレードする必要があります。
 - **バージョン 1 GISO からバージョン 2 GISO へのリリースでのシステムアップグレード** : 両方の GISO イメージの基本バージョンは同じでラベルが異なる場合、**install add** および **install activate** コマンドは同じバージョンの 2 つのイメージをサポートしません。その代わりに、**install source** コマンドを使用してデルタ RPM のみをインストールします。システムのリロードはデルタ RPM の再起動タイプに基づいています。
- install replace** コマンドを使用すると、ISO と既存のバージョン間の違いに関係なく、システムのリロードが実行されます。

- バージョン 1 GISO からバージョン 2 GISO へのリリース間でのシステム アップグレード：両方の GISO イメージの基本バージョンが異なります。install add および install activate コマンド、または install replace コマンドを使用して、システムアップグレードを実行します。ルータは、バージョン 2 GISO イメージを使用したアップグレード後にリロードされます。

ステップ 2 システム管理モードで **show install repository all** コマンドを実行し、ホスト、システム管理、および XR の RPM と基本 ISO を表示します。

```

sysadmin-vm:0_RP0#show install repository all
Admin repository
-----
asr9k-sysadmin-6.1.1

asr9k-sysadmin-hostos-6.1.1-r611.CSCcv10001.admin.x86_64
asr9k-sysadmin-system-6.1.1-r611.CSCcv10005.x86_64
....

XR repository
-----
asr9k-iosxr-mgbl-3.0.0.0-r611.x86_64
asr9k-xr-6.1.1
....

Host repository
-----
host-6.1.1

```

ステップ 3 **show install package <golden-iso>** コマンドを実行し、RPM のリストおよび GISO に組み込まれているパッケージを表示します。

- (注) GISO の RPM を一覧表示するには、GISO がインストールリポジトリに存在する必要があります。

```

Router#show install package asr9k-goldenk9-x64-6.1.3

Sun Dec  4 13:52:48.279 UTC
This may take a while ...
ISO Name: asr9k-goldenk9-x64-6.1.3
ISO Type: bundle
ISO Bundled: asr9k-mini-x64-6.1.3
Golden ISO Label: temp
ISO Contents:
ISO Name: asr9k-xr-6.1.3
ISO Type: xr
rpms in xr ISO:
  iosxr-os-asr9k-64-5.0.0.0-r613
  iosxr-ce-asr9k-64-3.0.0.0-r613
  iosxr-infra-asr9k-64-4.0.0.0-r613
  iosxr-fwding-asr9k-64-4.0.0.0-r613
  iosxr-routing-asr9k-64-3.1.0.0-r613
  ...

ISO Name: asr9k-sysadmin-6.1.3
ISO Type: sysadmin
rpms in sysadmin ISO:
  asr9k-sysadmin-topo-6.1.3-r613
  asr9k-sysadmin-shared-6.1.3-r613
  asr9k-sysadmin-system-6.1.3-r613
  asr9k-sysadmin-hostos-6.1.3-r613.admin
  ...

```

```

ISO Name: host-6.1.3
ISO Type: host
rpms in host ISO:
  asr9k-sysadmin-hostos-6.1.3-r613.host

Golden ISO Rpms:
  xr rpms in golden ISO:
    asr9k-k9sec-x64-2.2.0.1-r613.CSCxr33333.x86_64.rpm
    openssh-scp-6.6p1.p1-r0.0.CSCtp12345.xr.x86_64.rpm
    openssh-scp-6.6p1-r0.0.xr.x86_64.rpm
    asr9k-mp1s-x64-2.1.0.0-r613.x86_64.rpm
    asr9k-k9sec-x64-2.2.0.0-r613.x86_64.rpm

  sysadmin rpms in golden ISO:
    asr9k-sysadmin-system-6.1.3-r613.CSCcv11111.x86_64.rpm
    openssh-scp-6.6p1-r0.0.admin.x86_64.rpm
    openssh-scp-6.6p1.p1-r0.0.CSCtp12345.admin.x86_64.rpm

  host rpms in golden ISO:
    openssh-scp-6.6p1-r0.0.host.x86_64.rpm
    openssh-scp-6.6p1.p1-r0.0.CSCtp12345.host.x86_64.rpm
  
```

GISO 内の ISO、SMU、およびパッケージがルータにインストールされます。



第 9 章

クラシック ZTP を使用したルータの展開

大規模な環境でネットワークデバイスを手動で展開するには、熟練の作業者が必要であり、時間がかかります。

ゼロタッチプロビジョニング (ZTP) を使用すると、何千ものネットワークデバイスを数分以内にシームレスかつ正確にプロビジョニングでき、手動による介入は不要です。ZTP は、シェールまたは Python を使用し、構成ファイルやスクリプトを使用して簡単に定義できます。

ZTP には、次のような複数のオプションがあります。

- 大規模な環境で特定の設定を自動的に適用する。
- 特定の IOS XR イメージをダウンロードしてインストールする。
- 特定のアプリケーションパッケージやサードパーティアプリケーションを自動的にインストールする。
- 手動による介入なしでコンテナを展開する。
- 何千ものネットワークデバイスでソフトウェアバージョンを一度に簡単にアップグレードまたはダウングレードする。

ZTP を使用する利点

ZTP を使用すると、大規模なサービスプロバイダーのインフラストラクチャを簡単に管理できます。ZTP を使用する追加の利点は次のとおりです。

- ZTP を使用すると、ネットワーク内の任意の場所にルータをリモートでプロビジョニングできるため、ネットワークデバイスを展開するために専門家を派遣する必要がなく、IT コストが削減されます。
- ZTP を使用した自動プロビジョニングにより、遅延がなくなり、精度が向上するため、コスト効率が高くなり、カスタマーエクスペリエンスが向上します。

ZTP は、反復作業を自動化することで、ネットワーク管理者がより重要な作業に集中できるようにします。

- ZTP プロセスは、サービスの迅速な復元に役立ちます。問題を手動でトラブルシューティングする代わりに、システムを既知の動作ステータスにリセットできます。

ユースケース

次に、ZTP の便利なユースケースの一部を示します。

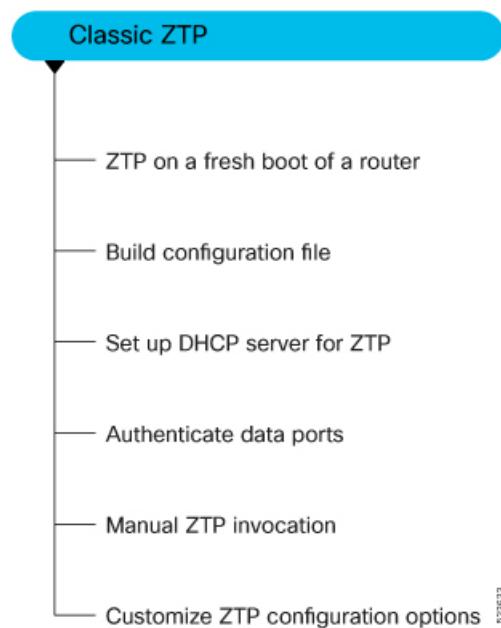
- ZTP を使用した Chef のインストール
- ZTP を使用した IOS XR と NSO の統合
- ZTP を使用した Puppet のインストール

ZTP は次のいずれかの方法で開始できます。

- **フレッシュブート**：設定が事前にロードされていないデバイスには、この方法を使用します。「ルータのフレッシュブート時の ZTP を使用したスタートアップガイド」を参照してください。[ルータのフレッシュブート時のゼロタッチプロビジョニング \(115 ページ\)](#) を参照してください
- **手動呼び出し**：完全に設定されたデバイスで ZTP を強制的に開始する場合は、この方法を使用します。[手動による ZTP の呼び出し \(116 ページ\)](#) を参照してください。

次の図は、クラシック ZTP を設定するために実行するタスクを示しています。

図 7:クラシック ZTP を設定するためのワークフロー



- [構成ファイルの作成 \(101 ページ\)](#)
- [データポートでの認証 \(109 ページ\)](#)
- [DHCP サーバーのセットアップ \(110 ページ\)](#)
- [ZTP 初期化ファイルのカスタマイズ \(113 ページ\)](#)
- [ルータのフレッシュブート時のゼロタッチプロビジョニング \(115 ページ\)](#)
- [手動による ZTP の呼び出し \(116 ページ\)](#)

構成ファイルの作成

ビジネスニーズに基づいて、構成ファイルやスクリプトファイルを使用して ZTP プロセスを開始できます。



- (注) ZTP のソースとして USB フラッシュドライブを使用する場合、スクリプトファイルはプロビジョニングに使用できません。スクリプトファイルは、USB フェッチャではサポートされていません。フェッチャでは、ztp.ini ファイルで定義されているプロビジョニングの詳細を取得するために ZTP プロセスで使用されるポートを定義します。

構成ファイルの内容は `!! IOS XR` で始まり、スクリプトファイルの内容は `#!/bin/bash`、`#!/bin/sh` または `#!/usr/bin/python` で始まります。

コンフィギュレーションファイルを作成したら、ztp_helper 関数 `xrapply` を使用してデバイスに適用します。

次に、構成ファイルの例を示します。

```
!! IOS XR
username root
group root-lr
password 0 lablab
!

hostname ios
alias exec al show alarms brief system active

interface HundredGigE 0/0/0/24
ipv4 address 10.10.10.55 255.255.255.0
no shutdown
!
```

ユーザースクリプトの作成

IOS XR bash シェルで実行されるユーザースクリプトやバイナリは、IOS XR CLI と対話して、設定、設定状態の確認を行い、オペレータが選択したワークフローに基づいて `exec` コマンドを実行するために使用できます。

シェルか Python を使用して ZTP スクリプトを作成します。ZTP には、ユーザースクリプト内で使用できる一連の CLI コマンドおよびシェルユーティリティが含まれています。



- (注) すでにプロビジョニングされているルータでは API を実行しないことを推奨します。ZTP ユーティリティ API は、ルータの初回起動時に ZTP スクリプトから実行されるように設計されています。API によって追加の操作が実行され、ブートプロセス中に要求されたアクションが実行され、アクションを実行する前に既存の設定に変更が加えられます。

ZTP ユーティリティ API には、ZTP ユーティリティ API を実行する前に ZTP ワークフローで実行される前提条件があります。それらの前提条件は、ブートプロセス中に特定のアクションを実行し、必要な設定の変更を行うのに役立ちます。

ZTP スクリプトの範囲外で ZTP ユーティリティを使用しないことを推奨します。このスクリプトの API では、すべてのアクションで `ztp` または `ztp-user` がユーザー名として使用されます。ZTP スクリプトの範囲外で実行される ZTP ユーティリティは、ZTP ワークフローから実行されないため、失敗する可能性があるため、デバイスの設定が変更され、他の関連する操作に影響を与える可能性があります。ZTP ユーティリティが ZTP スクリプトの範囲外で実行された場合、ユーザー名 `ztp` または `ztp-user` を使用してスクリプトが実行されたことがログに表示されるため、スクリプトがワークフローから実行されたと誤解されます。

ZTP シェルユーティリティ

ZTP には、ユーザー スクリプト内から送信可能な一連のシェルユーティリティが含まれています。`ztp_helper.sh` は、ユーザー スクリプトによって送信可能なシェル スクリプトです。`ztp_helper.sh` は、一部の XR 機能にアクセスするための単純なユーティリティを提供します。次の `bash` 関数を呼び出すことができます。

- **xrcmd** : 単一の XR `exec` コマンド `xrcmd "show running"` を実行するために使用されます。
- **xrapplly** : ファイルに指定された構成ブロックを適用します。

```
cat >/tmp/config <<%%
!! XR config example
hostname nodel-mgmt-via-xrapplly
%%
xrapplly /tmp/config
```

- **xrapplly_with_reason** : XR 構成のブロックをロギング目的の理由とともに適用するために使用されます。

```
cat >/tmp/config <<%%
!! XR config example
hostname nodel-mgmt-via-xrapplly
%%
xrapplly_with_reason "this is a system upgrade" /tmp/config
```

- **xrapplly_string** : XR 構成のブロックを 1 行で適用するために使用されます。

```
xrapplly_string "hostname foo\interface HundredGigE0/0/0/24\nipiv4 address 1.2.3.44
255.255.255.0\n"
```

- **xrapply_string_with_reason** : ログ目的の理由とともに XR 構成のブロックを 1 行で適用するために使用されます。

```
xrapply_string_with_reason "system renamed again" "hostname venus\n interface
HundredGigE0/0/0/24\n
ipv4 address 172.30.0.144/24\n"
```

- **xrreplace** : XR 構成の置換をファイルを介して XR 名前空間に適用するために使用されま

```
cat rtr.cfg <<%%
!! XR config example
hostname node1-mgmt-via-xrreplace
%%
xrreplace rtr.cfg
```

- **xrapply_with_extra_auth** : 認証を必要とする XR 構成をファイルを介して XR 名前空間に適用するために使用されます。 **xrapply_with_extra_auth** API が、エイリアス、フレックスグループなどの追加の認証を必要とする構成が適用されるときに使用されます。この API は、追加の権限を取得するために、内部で認証と承認を実行します。

```
cat >/tmp/config <<%%
!! XR config example
alias exec alarms show alarms brief system active
alias exec version run cat /etc/show_version.txt
%%
xrapply_with_extra_auth >/tmp/config
```

- **xrreplace_with_extra_auth** : XR 構成の置換をファイルを介して XR 名前空間に適用するために使用されます。 **xrreplace_with_extra_auth** API は、エイリアス、Flex グループなど、追加の認証を必要とする構成が適用されるときに使用されます。この API は、追加の権限を取得するために、内部で認証と承認を実行します。

```
cat >/tmp/config <<%%
!! XR config example
alias exec alarms show alarms brief system active
alias exec version run cat /etc/show_version.txt
%%
xrreplace_with_extra_auth >/tmp/config
```

API 実装の動作



(注) **xrcmd**、**xrapply**、および **xrreplace** API またはユーティリティでは、特定のアクションを実行するための一連の内部操作が実行されます。順番に実行される内部操作には、次の操作が含まれます。

- ユーザーの作成：この操作では、他の操作を実行する前に `ztp-user`（一時的なユーザー）が生成されます。
- コマンドの実行または設定の適用：この操作には、コマンドの実行、パーサーユーティリティを使用した設定の適用、または `cfg-mgr` による設定の適用が含まれます。
- ユーザーの削除：この操作では、XR 設定から `ztp-user`（一時的なユーザー）が削除されます。

これらの内部操作に加えて、**xrapply_with_extra_auth** および **xrreplace_with_extra_auth** API では、設定を適用する前に認証プロセスが実行されます。

ZTP ヘルパー Python ライブラリ

ZTP Python ライブラリでは、`ZtpHelpers` と呼ばれる単一の Python クラスを定義します。ヘルパースクリプトは `/pkg/bin/ztp_helper.sh` にあります。

ZtpHelpers クラスのメソッド

次に、`ZtpHelpers` クラスのユーティリティメソッドを示します。

- `init(self, syslog_server=None, syslog_port=None, syslog_file=None):`

```
__init__ constructor
:param syslog_server: IP address of reachable Syslog Server
:param syslog_port: Port for the reachable syslog server
:param syslog_file: Alternative or addon file for syslog
:type syslog_server: str
:type syslog_port: int
:type syslog_file: str
```

All parameters are optional. When nothing is specified during object creation, then all logs are sent to a log rotated file `/tmp/ztp_python.log` (max size of 1MB).

- `setns(cls, fd, nstype):`

```
Class Method for setting the network namespace
:param cls: Reference to the class ZtpHelpers
:param fd: incoming file descriptor
:param nstype: namespace type for the sentns call
:type nstype: int
    0 Allow any type of namespace to be joined.
    CLONE_NEWNET = 0x40000000 (since Linux 3.0)
    fd must refer to a network namespace
```

- `get_netns_path(cls, nspath=None, nsname=None, nspid=None):`

```

Class Method to fetch the network namespace filepath
associated with a PID or name
:param cls: Reference to the class ZtpHelpers
:param nspath: optional network namespace associated name
:param nspid: optional network namespace associate PID
:type nspath: str
:type nspid: int
:return: Return the complete file path
:rtype: str

• toggle_debug(self, enable):

Enable/disable debug logging
:param enable: Enable/Disable flag
:type enable: int

• set_vrf(self, vrfname=None):

Set the VRF (network namespace)
:param vrfname: Network namespace name
corresponding to XR VRF

• download_file(self, file_url, destination_folder):

Download a file from the specified URL
:param file_url: Complete URL to download file
:param destination_folder: Folder to store the
downloaded file

:type file_url: str
:type destination_folder: str
:return: Dictionary specifying download success/failure
Failure => { 'status' : 'error' }
Success => { 'status' : 'success',
'filename' : 'Name of downloaded file',
'folder' : 'Directory location of downloaded file'}

:rtype: dict

• setup_syslog(self):

正しい VRF (ネットワーク名前空間) で sysloghandler を正しく設定し、リモート syslog
サーバー、ローカルファイル、またはデフォルトの log-rotated ログファイルを指すメソッ
ド。

• xrcmd(self, cmd=None):

Issue an IOS-XR exec command and obtain the output
:param cmd: Dictionary representing the XR exec cmd
and response to potential prompts
{ 'exec_cmd': '', 'prompt_response': '' }
:type cmd: dict
:return: Return a dictionary with status and output
{ 'status': 'error/success', 'output': '' }
:rtype: dict

• xraply(self, filename=None, reason=None):

Apply Configuration to XR using a file
:param file: Filepath for a config file
with the following structure:
!
XR config command
!
end

```

```

:param reason: Reason for the config commit.
                Will show up in the output of:
                "show configuration commit list detail"
:type filename: str
:type reason: str
:return: Dictionary specifying the effect of the config change
        { 'status' : 'error/success', 'output': 'exec command based on
status'}
                In case of Error: 'output' = 'show configuration failed'
                In case of Success: 'output' = 'show configuration commit changes
last 1'
:rtype: dict

• xrapplly_string(self, cmd=None, reason=None):
Apply Configuration to XR using a single line string
:param cmd: Single line string representing an XR config command
:param reason: Reason for the config commit.
                Will show up in the output of:
                "show configuration commit list detail"
:type cmd: str
:type reason: str
:return: Dictionary specifying the effect of the config change
        { 'status' : 'error/success', 'output': 'exec command based on
status'}
                In case of Error: 'output' = 'show configuration failed'
                In case of Success: 'output' = 'show configuration commit changes
last 1'
:rtype: dict

• xrreplace(self, filename=None):
Replace XR Configuration using a file

:param file: Filepath for a config file
                with the following structure:

                !
                XR config commands
                !
                end
:type filename: str
:return: Dictionary specifying the effect of the config change
        { 'status' : 'error/success', 'output': 'exec command based on
status'}
                In case of Error: 'output' = 'show configuration failed'
                In case of Success: 'output' = 'show configuration commit changes
last 1'
:rtype: dict

```


API 実装の動作



(注) **xrcmd**、**xrapply**、および **xrreplace** API またはユーティリティでは、特定のアクションを実行するための一連の内部操作が実行されます。順番に実行される内部操作には、次の操作が含まれます。

- ユーザーの作成：この操作では、他の操作を実行する前に `ztp-user`（一時的なユーザー）が生成されます。
- コマンドの実行または設定の適用：この操作には、コマンドの実行、パーサーユーティリティを使用した設定の適用、または `cfg-mgr` による設定の適用が含まれます。
- ユーザーの削除：この操作では、XR 設定から `ztp-user`（一時的なユーザー）が削除されます。

例

次に、Python のサンプルスクリプトを示します。

```
[apple2:~]$ python sample_ztp_script.py

##### Debugs enabled #####

##### Change context to user specified VRF #####

##### Using Child class method, setting the root user #####

2016-12-17 04:23:24,091 - DebugZTPLogger - DEBUG - Config File content to be applied !
    username netops
    group root-lr
    group cisco-support
    secret 5 $1$7kTu$zjrgqbgW08vEXsYzUycXw1
    !
    end
2016-12-17 04:23:28,546 - DebugZTPLogger - DEBUG - Received exec command request: "show
    configuration commit changes last 1"
2016-12-17 04:23:28,546 - DebugZTPLogger - DEBUG - Response to any expected prompt ""
Building configuration...
2016-12-17 04:23:29,329 - DebugZTPLogger - DEBUG - Exec command output is [!!! IOS XR
Configuration version = 6.2.1.21I', 'username netops', 'group root-lr', 'group
cisco-support', 'secret 5 $1$7kTu$zjrgqbgW08vEXsYzUycXw1', '!', 'end']
2016-12-17 04:23:29,330 - DebugZTPLogger - DEBUG - Config apply through file successful,
last change = [!!! IOS XR Configuration version = 6.2.1.21I', 'username netops', 'group
root-lr', 'group cisco-support', 'secret 5 $1$7kTu$zjrgqbgW08vEXsYzUycXw1', '!', 'end']

##### Debugs Disabled #####

##### Executing a show command #####

Building configuration...
{'output': [!!! IOS XR Configuration version = 6.2.1.21I',
            '!!! Last configuration change at Sat Dec 17 04:23:25 2016 by UNKNOWN',
```

```

        '!',
        'hostname customer2',
        'username root',
        'group root-lr',
        'group cisco-support',
        'secret 5 $1$7kTu$zjrgqbgW08vEXsYzUycXw1',
        '!',
        'username noc',
        'group root-lr',
        'group cisco-support',
        'secret 5 $1$7kTu$zjrgqbgW08vEXsYzUycXw1',
        '!',
        'username netops',
        'group root-lr',
        'group cisco-support',
        'secret 5 $1$7kTu$zjrgqbgW08vEXsYzUycXw1',
        '!',
        'username netops2',
        'group root-lr',
        'group cisco-support',
        'secret 5 $1$7kTu$zjrgqbgW08vEXsYzUycXw1',
        '!',
        'username netops3',
        'group root-lr',
        'group cisco-support',
        'secret 5 $1$7kTu$zjrgqbgW08vEXsYzUycXw1',
        '!',
        'cdp',
        'service cli interactive disable',
        'interface MgmtEth0/RP0/CPU0/0',
        'ipv4 address 11.11.11.59 255.255.255.0',
        '!',
        'interface TenGigE0/0/0/24',
        'shutdown',
        '!',
        'interface TenGigE0/0/0/25',
        'shutdown',
        '!',

        'router static',
        'address-family ipv4 unicast',
        '0.0.0.0/0 11.11.11.2',
        '!',
        '!',
        'end'],
    'status': 'success'}

##### Apply valid configuration using a file #####

Building configuration...
{'status': 'success', 'output': ['!! IOS XR Configuration version = 6.2.1.21I', 'hostname
customer', 'cdp', 'end']}

##### Apply valid configuration using a string #####

Building configuration...
{'output': ['!! IOS XR Configuration version = 6.2.1.21I',
            'hostname customer2',
            'end'],
 'status': 'success'}

##### Apply invalid configuration using a string #####

{'output': ['!! SYNTAX/AUTHORIZATION ERRORS: This configuration failed due to',

```

```
'!! one or more of the following reasons:',
'!! - the entered commands do not exist,',
'!! - the entered commands have errors in their syntax,',
'!! - the software packages containing the commands are not active,'
```

ヘルパー API の詳細については、<https://github.com/ios-xr/iosxr-ztp-python#iosxr-ztp-python> を参照してください。

データポートでの認証

新規起動時は ZTP プロセスが管理ポートから開始されますが、データポートに切り替えることができます。DHCP サーバーとの接続を検証するために、DHCP オプション 43 (IPv4)、およびオプション 17 (IPv6) を使用してデータポートで認証が実行されます。これらの DHCP オプションはオプションスペースで定義され、**dhcpd.conf** および **dhcpd6.conf** コンフィギュレーションファイルに含まれています。オプションスペースを定義する際は、認証に次のパラメータを指定する必要があります。

- 認証コード：認証コードは 0 または 1 のいずれかです。0 は認証が不要であることを示し、1 は MD5 チェックサムが必要であることを示します。



(注) IPv4 のオプション 43 と IPv6 のオプション 17 が無効になっている場合、認証は失敗します。

- クライアント ID：クライアント ID は「xr-config」にする必要があります。
- MD5 チェックサム：これはシャーマシのシリアル番号です。MD5 チェックサムは、**echo -n \$SERIALNUMBER | md5sum | awk '{print \$1}'** を使用して取得できます。

dhcpd.conf の設定例を次に示します。以下の例では、**VendorInfo** というオプションスペースが、認証用の 3 つのパラメータを使用して定義されています。

```
class "vendor-classes" {
    match option vendor-class-identifier;
}

option space VendorInfo;
option VendorInfo.clientId code 1 = string;
option VendorInfo.authCode code 2 = unsigned integer 8;
option VendorInfo.md5sum code 3 = string;
option vendor-specific code 43 = encapsulate VendorInfo;
subnet 10.65.2.0 netmask 255.255.255.0 {
    option subnet-mask 255.255.255.0;
    option routers 10.65.2.1;
    range 10.65.2.1 10.65.2.200;
}
host cisco-mgmt {
    hardware ethernet 00:50:60:45:67:01;
    fixed-address 10.65.2.39;
    vendor-option-space VendorInfo;
    option VendorInfo.clientId "xr-config";
    option VendorInfo.authCode 1;
    option VendorInfo.md5sum "aedf5c457c36390c664f5942ac1ae3829";
```

```
option bootfile-name "http://10.65.2.1:8800/admin-cmd.sh";
}
```

dhcpd6.conf コンフィギュレーション ファイルの例を次に示します。以下の例では、コード幅 2 および縦の長さ 2 (IPv6 では DHCP 標準に従う) の **VendorInfo** というオプション スペースが、認証用の 3 つのパラメータを使用して定義されています。

```
log-facility local7;
option dhcp6.name-servers 2001:1451:c632:1::1;
option dhcp6.domain-search "cisco.com";
dhcpv6-lease-file-name "/var/lib/dhcpd/dhcpd6.leases";
option dhcp6.info-refresh-time 21600;
option dhcp6.bootfile-url code 59 = string;
option dhcp6.user-class code 15 = string;
option space CISCO-XR-CONFIG code width 2 length width 2;
option CISCO-XR-CONFIG.client-identifier code 1 = string;
option CISCO-XR-CONFIG.authCode code 2 = integer 8;
option CISCO-XR-CONFIG.md5sum code 3 = string;
option vsio.CISCO-XR-CONFIG code 9 = encapsulate CISCO-XR-CONFIG;
subnet6 2001:1451:c632:1::/64{
  range6 2001:1451:c632:1::2 2001:1451:c632:1::9;
  option CISCO-XR-CONFIG.client-identifier "xr-config";
  option CISCO-XR-CONFIG.authCode 1;
  #valid md5
  option CISCO-XR-CONFIG.md5sum "90fd845ac82c77f834d57a034658d0f0";
  if option dhcp6.user-class = 00:04:69:50:58:45 {
    option dhcp6.bootfile-url "http://[2001:1851:c632:1::1]/cisco-2/image.iso";
  }
  else {
    #option dhcp6.bootfile-url "http://[2001:1851:c632:1::1]/cisco-2/cisco-mini-x.iso.sh";

    option dhcp6.bootfile-url "http://[2001:1851:c632:1::1]/cisco-2/ztp.cfg";
  }
}
```

DHCP サーバーのセットアップ

ZTP を使用して有効な IPv4 または IPv6 アドレスを動作させるには、DHCP サーバーから構成スクリプトにポインタが送信される必要があります。

ルータからの DHCP 要求には、ルータ自身を識別するための次の DHCP オプションがあります。

- **オプション 60** : 「vendor-class-identifier」 : 次の 4 つの要素を識別するために使用されません。
 - クライアントのタイプ (例 : PXEClient)
 - システム (Arch) のアーキテクチャ : 例 : 00009 x86-64 CPU を使用した EFI システムの特定
 - Universal Network Driver Interface (UNDI) :
 - 例 : 003010 (最初の 3 オクテットはメジャーバージョンを特定し、最後の 3 オクテットはマイナーバージョンを特定)
 - 製品 ID (PID) :

- **オプション 61** : 「dhcp-client-identifier」 : デバイスのシリアル番号を識別するために使用されます。
- **オプション 66** : TFTP サーバー名を要求するために使用されます。
- **オプション 67** : TFTP ファイル名を要求するために使用されます。
- **オプション 97** 「uuid」 : 汎用一意識別子を 128 ビット値で識別するために使用されます (現時点では使用できません) 。

例

次の DHCP 要求の例は、固定 IP アドレスと、管理インターフェイスの MAC アドレスを含む構成ファイルを示しています。

```
host cisco-rp0 {
  hardware ethernet e4:c7:22:be:10:ba;
  fixed-address 172.30.12.54;
  filename "http://172.30.0.22/configs/cisco-1.config";
}
```

次の DHCP 要求の例は、固定 IP アドレスと、iPXE ("xr-config" オプション) を使用してシステムを再イメージ化する機能とともに、管理インターフェイスの MAC アドレスを含む構成ファイルを示しています。

```
host cisco-rp0 {
  hardware ethernet e4:c7:22:be:10:ba;
  fixed-address 172.30.12.54;
  if exists user-class and option user-class = "iPXE" {
    filename = "http://172.30.0.22/boot.ipxe";
  } elseif exists user-class and option user-class = "xr-config" {
    filename = "http://172.30.0.22/scripts/cisco-rp0_ztp.sh";
  }
}
```

DHCP サーバーはデバイスを識別し、ファイル名オプションとして IOS XR 構成ファイルまたは ZTP スクリプトのいずれかを使用して応答します。

DHCP サーバーは、次の DHCP オプションを使用して応答します。

- BOOTP ファイル名を使用してスクリプトや構成の場所を指定する DHCPv4。
- オプション 67 (bootfile-name) を使用してスクリプトや構成の場所を指定する DHCPv4。
- オプション 15 を使用する DHCPv6 : サーバーに ZTP 要求を識別させるためにこのオプションを設定した場合は、Linux または ISC サーバーのサーバー設定を更新してください。ZTP のユーザークラスを確認するために必要なサーバー側の設定例を、次の例に示します。

```
if exists dhcp6.user-class and (substring(option dhcp6.user-class, 0, 9) = "xr-config"
or substring(option dhcp6.user-class, 2, 9) = "xr-config"){
  #
}
```

- オプション 59 (OPT_BOOTFILE_URL) を使用してスクリプトや構成の場所を指定する DHCPv6

次の例は、bootfile-name (オプション 67) を使用した DHCP 応答を示しています。

```

option space cisco-vendor-id-vendor-class code width 1 length width 1;
option vendor-class.cisco-vendor-id-vendor-class code 9 = {string};

##### Network 11.11.11.0/24 #####
shared-network 11-11-11-0 {

##### Pools #####
  subnet 11.11.11.0 netmask 255.255.255.0 {
    option subnet-mask 255.255.255.0;
    option broadcast-address 11.11.11.255;
    option routers 11.11.11.2;
    option domain-name-servers 11.11.11.2;
    option domain-name "cisco.local";
    # DDNS statements
    ddns-domainname "cisco.local.";
    # use this domain name to update A RR (forward map)
    ddns-rev-domainname "in-addr.arpa.";
    # use this domain name to update PTR RR (reverse map)

  }

##### Matching Classes #####

  class "cisco" {
    match if (substring(option dhcp-client-identifier,0,11) = "FGE194714QS");
  }

  pool {
    allow members of "cisco";
    range 11.11.11.47 11.11.11.50;
    next-server 11.11.11.2;

    if exists user-class and option user-class = "iPXE" {
      filename="http://11.11.11.2:9090/cisco-mini-x-6.2.25.10I.iso";
    }

    if exists user-class and option user-class = "xr-config"
    {
      if (substring(option vendor-class.cisco-vendor-id-vendor-class,19,99)="cisco")
      {
        option bootfile-name "http://11.11.11.2:9090/scripts/exhaustive_ztp_script.py";
      }
    }

    ddns-hostname "cisco-local";
    option routers 11.11.11.2;
  }
}

```



重要 システムで Cisco IOS XR リリース 7.3.1 以前を実行している場合、**user-class = "exr-config"** を送信するデバイスは受け入れられますが、Cisco IOS XR リリース 7.3.2 以降の場合は、**user-class = "xr-config"** のみを使用する必要があります。

Cisco IOS XR リリース 7.3.2 以降では、以下を使用します。

```
host cisco-rp0 {
  hardware ethernet e4:c7:22:be:10:ba;
  fixed-address 172.30.12.54;
  if exists user-class and option user-class = "iPXE" {
    filename = "http://172.30.0.22/boot.ipxe";
  } elseif exists user-class and option user-class = "xr-config" {
    filename = "http://172.30.0.22/scripts/cisco-rp0_ztp.sh";
  }
}
```

また、Cisco IOS XR リリース 7.3.1 以前のリリースから Cisco IOS XR リリース 7.3.2 以降のリリースにアップグレードする場合は、以下を使用します。

```
host cisco-rp0 {
  hardware ethernet e4:c7:22:be:10:ba;
  fixed-address 172.30.12.54;
  if exists user-class and option user-class = "iPXE" {
    filename = "http://172.30.0.22/boot.ipxe";
  } elseif exists user-class and option user-class = "exr-config" {
    filename = "http://172.30.0.22/scripts/cisco-rp0_ztp.sh";
  }
}
```

ZTP 初期化ファイルのカスタマイズ

ztp.ini ファイルでは、以下の ZTP の設定可能オプションをカスタマイズできます。

- [ZTP] : 起動時に CLI を使用するか、*ztp.ini* ファイルを編集することで、ZTP を有効または無効にできます。
- [Retry] : ZTP DHCP 再試行メカニズムを設定します。設定可能な値は *infinite* と *once* です。
- [Fetcher Priority] : フェッチャーでは、プロビジョニングの詳細を取得するために ZTP が使用するポートを定義します。デフォルトでは、各ポートにおけるフェッチャーの優先順位は *ztp.ini* ファイルで定義されています。フェッチャーのデフォルトの優先順位は変更できます。許可された範囲は 0 ~ 10 です。



(注) 数字が小さいほど、優先順位が高くなります。値 0 は優先順位が最も高く、10 は優先順位が最も低くなります。

次の例では、Mgmt4 ポートの優先順位が最も高くなります。

```
[Fetcher Priority]
Mgmt4: 0
```

```
Mgmt6: 1
DPort4: 2
DPort6: 3
```

- `progress_bar` : コンソールのプログレスバーを有効にします。デフォルトでは、プログレスバーは無効になっています。プログレスバーを有効にするには、`ztp.ini` ファイルに次のエントリを追加します。

```
[Options]
progress_bar: True
```

デフォルトでは、`ztp.ini` ファイルは `/pkg/etc/` にあります。ZTP の設定可能オプションを変更するには、`/disk0:/ztp/` ディレクトリにファイルのコピーを作成し、`ztp.ini` ファイルを編集します。

デフォルトのオプションにリセットするには、`/disk0:/ztp/` ディレクトリにある `ztp.ini` ファイルを削除します。



(注) インストール中の問題を回避するために、`/pkg/etc/` にある `ztp.ini` ファイルを編集または削除しないでください。

次に、`ztp.ini` ファイルの例を示します。

```
[Startup]
start: True
retry_forever: True

[Fetcher Priority]
Mgmt4: 1
Mgmt6: 2
DPort4: 3
DPort6: 4
```

CLI を使用した ZTP の有効化

CLI を使用して ZTP を有効にする場合は、`ztp enable` コマンドを使用します。

設定例

```
Router#ztp enable
Fri Jul 12 16:09:02.154 UTC
Enable ZTP? [confirm] [y/n] :y
ZTP Enabled.
```

CLI を使用した ZTP の無効化

CLI を使用して ZTP を無効にする場合は、`ztp disable` コマンドを使用します。

設定例

```
Router#ztp disable
Fri Jul 12 16:07:18.491 UTC
Disable ZTP? [confirm] [y/n] :y
ZTP Disabled.
Run ZTP enable to run ZTP again.
```

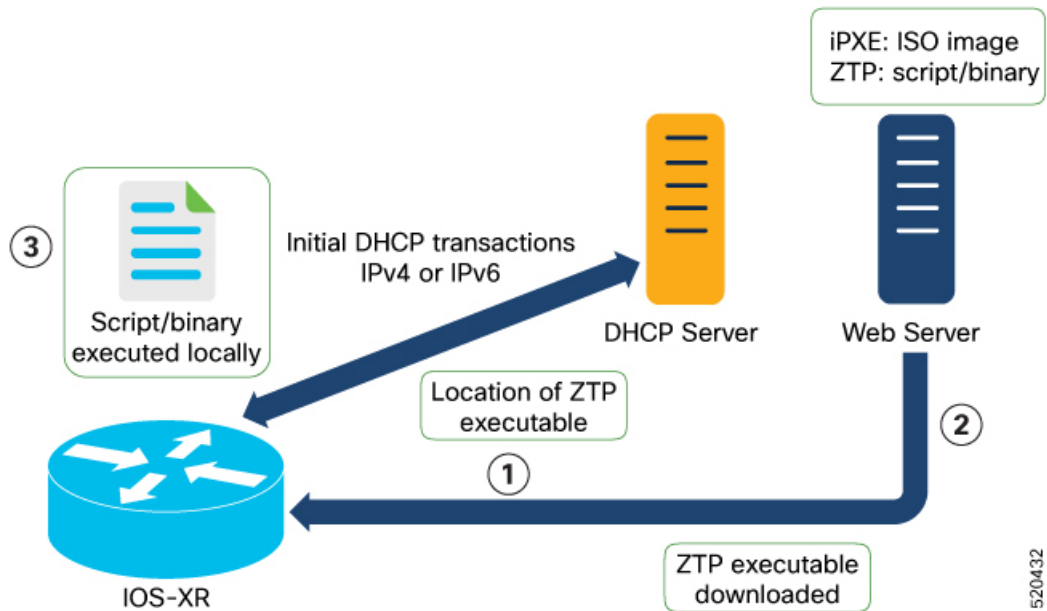

ルータのフレッシュブート時のゼロタッチプロビジョニング

事前に設定されていないデバイスを起動する場合、ZTP プロセスが自動的に開始されます。

DHCP を使用したフレッシュブート

事前に設定されていないデバイスを起動する場合、ZTP プロセスが自動的に開始されます。ZTP プロセス中に、ルータは DHCP サーバーから構成ファイルの詳細を受信します。

次の図は、ZTP プロセスのワークフローの概要を示しています。



IOS XR イメージを使用してネットワークデバイスを起動すると、ZTP プロセスが開始されます。ZTP プロセスは、事前に設定されていないデバイスでのみ開始されます。

フレッシュブート用の ZTP プロセスのワークフローの概要を次に示します。

1. ZTP が、ZTP 構成ファイルまたはユーザスクリプトを取得するための DHCP 要求を送信します。ブートストラップサーバーがデバイスを一意に識別できるように、ZTP で以下の DHCP オプションが送信されます。

- DHCP (v4/v6) client-id : シリアル番号
- DHCPv4 オプション 124 : ベンダー、プラットフォーム、シリアル番号
- DHCPv6 オプション 16 : ベンダー、プラットフォーム、シリアル番号

次に、ZTP プロセスのデフォルトのシーケンシャルフローを示します。

- 最初に、ZTP がすべての管理ポートで IPv4 DHCP 要求を送信します。障害が発生した場合、ZTP がすべての管理ポートで IPv6 DHCP 要求を送信します。
- 最初に、ZTP がすべてのデータポートで IPv4 DHCP 要求を送信します。障害が発生した場合、ZTP がすべてのデータポートで IPv6 DHCP 要求を送信します。

デフォルトのシーケンシャルフローは構成ファイルで定義されるため、その構成ファイルを使用してシーケンスを変更できます。

2. DHCP サーバーがデバイスを識別し、次のいずれかのオプションを使用して DHCP 応答で応答します。

DHCP サーバーは、DHCP オプションで応答するように設定する必要があります。

- BOOTP ファイル名を使用してスクリプトや構成の場所を指定する DHCPv4。
- オプション 67 (bootfile-name) を使用してスクリプトや構成の場所を指定する DHCPv4。
- オプション 59 (OPT_BOOTFILE_URL) を使用してスクリプトや構成の場所を指定する DHCPv6

3. ネットワークデバイスが、DHCP 応答で提供される URI の場所を使用して、Web サーバーからファイルをダウンロードします。
4. デバイスが、HTTP サーバーから構成ファイルまたはスクリプトファイルを受信します。



- (注)
- ダウンロードしたファイルの内容が !! IOS XR は構成ファイルと見なされます。
 - ダウンロードしたファイルの内容が #!/bin/bash、#!/bin/sh、または#!/usr/bin/python で始まる場合はスクリプトファイルと見なされます。

5. デバイスによって、構成ファイルが適用されるか、デフォルトの bash シェルでスクリプトまたはバイナリが実行されます。
6. これで、ネットワークデバイスが稼働しています。

手動による ZTP の呼び出し

コマンドライン インターフェイスを使用して、ゼロタッチプロビジョニング (ZTP) を手動で呼び出せます。この方法は、再起動せずに ZTP 設定を確認するのに最適です。この手動によるアプローチは、ルータを段階的にプロビジョニングするのに役立ちます。インターフェイス (データポートまたは管理ポート) 上で ZTP を呼び出す場合、最初にインターフェイスを起動して設定する必要はありません。

インターフェイスがダウンしている場合でも、ztp initiate コマンドを実行でき、ZTP スクリプトによってインターフェイスが起動され、dhclient が呼び出されるため、ZTP は、可用性に関係なくすべてのインターフェイスで実行できます。

ZTP コマンドを手動で呼び出し、すべてのインターフェイスで ZTP を強制的に実行するには、次のコマンドを使用します。

- **ztp initiate** : 新しい ZTP DHCP セッションを呼び出します。ログは `/disk0:/ztp/ztp.log` にあります。

設定例

```
Router#ztp initiate debug verbose interface HundredGigE 0/0/0/24
Invoke ZTP? (this may change your configuration) [confirm] [y/n] :
```

- **ztp terminate** : 進行中の ZTP セッションを終了します。

設定例

```
Router #ztp terminate verbose
Mon Oct 10 16:52:38.507 UTC
Terminate ZTP? (this may leave your system in a partially configured state) [confirm]
[y/n] :y
ZTP terminated
```

- **ztp enable** : 起動時に ZTP を有効にします。

設定例

```
Router#ztp enable
Fri Jul 12 16:09:02.154 UTC
Enable ZTP? [confirm] [y/n] :y
ZTP Enabled.
```

- **ztp disable** : 起動時に ZTP を無効にします。

設定例

```
Router#ztp disable
Fri Jul 12 16:07:18.491 UTC
Disable ZTP? [confirm] [y/n] :y
ZTP Disabled.
Run ZTP enable to run ZTP again.
```

- **ztp clean** : ZTP の状態ファイルのみを削除します。

設定例

```
Router#ztp clean verbose
Mon Oct 10 17:03:43.581 UTC
Remove all ZTP temporary files and logs? [confirm] [y/n] :y
All ZTP files have been removed.
If you now wish ZTP to run again from boot, do 'conf t/commit replace' followed by
reload.
```

ログファイル `ztp.log` は `/var/log` フォルダに保存され、ログファイルのコピーはソフトリンクを使用して `/disk0:/ztp/ztp.log` で使用できます。ただし、**ztp clean** を実行すると、現在の ZTP ログが保存されている `/var/log` フォルダではなく、ディスクに保存されているファイルがクリアされます。現在の ZTP からログを実行するには、`/var/log/` フォルダから ZTP ログ ファイルを手動でクリアする必要があります。

設定

このタスクでは、手動 ZTP 呼び出しの最も一般的なユースケース（ZTP の呼び出し）を示します。

1. 起動している、または起動可能なすべてのデータポートで DHCP セッションを呼び出します。ZTP はバックグラウンドで実行されます。進捗状況を確認するには、`show logging` を使用するか、`/disk0:/ztp/ztp.log` を確認します。

設定例

```
Router# ztp initiate dataport
```

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。