



## デバイスのセットアップ設定の実行

- [ソフトウェアインストールの制約事項 \(1 ページ\)](#)
- [デバイスセットアップ設定の実行に関する情報 \(1 ページ\)](#)
- [デバイスセットアップ設定の実行方法 \(14 ページ\)](#)
- [デバイスのセットアップ設定のモニターリング \(30 ページ\)](#)
- [デバイスのセットアップを実行する場合の設定例 \(35 ページ\)](#)
- [デバイスセットアップの実行に関する追加情報 \(50 ページ\)](#)
- [デバイスセットアップ設定の実行に関する機能履歴 \(50 ページ\)](#)

## ソフトウェアインストールの制約事項

- サブパッケージのインストールはサポートされていません。

## デバイスセットアップ設定の実行に関する情報

IP アドレスの割り当ておよび DHCP 自動設定を含む初期 device 設定タスクを実行する前に、このモジュールのセクションを確認します。

## デバイスブート プロセス

device を起動するには、ハードウェア設置ガイドの手順に従って device を設置して電源投入し、device の初期設定を行う必要があります。

通常の起動プロセスにはブートローダソフトウェアの動作が含まれ、以下のアクティビティが実行されます。

- 下位レベルの CPU 初期化を行います。CPU レジスタを初期化することにより、物理メモリがマッピングされる場所、容量、速度などを制御します。
- CPU サブシステムの電源投入時セルフテスト (POST) を実行し、システム DRAM をテストします。

- システム ボード上のファイル システムを初期化します。
- デフォルトのオペレーティング システム ソフトウェア イメージをメモリにロードし、**device**を起動します。

ブート ロードにより、オペレーティング システムがロードされる前に、ファイル システムにアクセスすることができます。ブート ロードの使用目的は通常、オペレーティング システムのロード、展開、および起動に限定されます。オペレーティング システムが CPU を制御できるようになると、ブートローダは、次にシステムがリセットされるか電源が投入されるまでは非アクティブになります。

**device**情報を割り当てるには、PC または端末をコンソール ポートに接続するか、PC をイーサネット管理ポートに接続して、PC または端末エミュレーション ソフトウェアのボーレートおよびキャラクタフォーマットを**device**のコンソールポートの設定と一致させておく必要があります。

- デフォルトのボーレートは 9600 です。
- デフォルトのデータ ビットは 8 です。



(注) データビットオプションを8に設定した場合、パリティオプションは「なし」に設定します。

- デフォルトのストップ ビットは 2 (マイナー) です。
- デフォルトのパリティ設定は「なし」です。

## ソフトウェア インストールの概要

ソフトウェア インストール機能では、イメージの完全インストール、ソフトウェア メンテナンスアップグレード (SMU)、インサービスソフトウェアアップグレード (ISSU)、およびインサービス モデルアップグレード (データ モデルパッケージ) など、さまざまなタイプのアップグレードを同じように実行できます。

ソフトウェア インストール機能は、インストール モードでソフトウェアを1つのバージョンから別のバージョンへと移行する際に役立ちます。**install** コマンドを特権 EXEC モードで使用して、ソフトウェアイメージをインストールまたはアップグレードします。また、インストール モードを使用して以前のバージョンのソフトウェア イメージにダウングレードすることもできます。

Cisco IOS XE ソフトウェアをアップグレードするために使用する方式は、スイッチが動作しているのがインストール モードかバンドル モードかによって異なります。バンドル モードまたは統合ブートモードでは、ローカルまたはリモートロケーションから **.bin image** ファイルを使用してデバイスをブートします。インストールブートモードでは、ブートローダが **packages.conf** ファイルを使用してデバイスをブートします。

スイッチでは、次のソフトウェア インストール機能がサポートされています。

- スタンドアロン スイッチでのソフトウェア バンドルのインストール。
- 以前にインストールしたパッケージセットへのソフトウェア ロールバック。
- 有効なインストール済みパッケージがブート フラッシュに存在しない場合の緊急インストール。

## ソフトウェアのブートモード

deviceでは、ソフトウェア パッケージを起動するための次の2種類のモードがサポートされています。

- インストール モード
- バンドル モード

### インストールモードでのブート

以下のフラッシュ内のソフトウェア パッケージのプロビジョニング ファイルを起動して、インストール モードでdeviceを起動できます:

```
Switch: boot flash:packages.conf
```



- 
- (注) 特定リリース用の `packages.conf` ファイルが「ソフトウェア パッケージのインストール」という項で説明するインストール ワークフローで作成されています。
- 

プロビジョニング ファイルには、起動、マウント、実行するソフトウェア パッケージのリストが含まれます。インストールされている各パッケージの ISO ファイル システムは、フラッシュからルート ファイル システムに直接マウントされます。



- 
- (注) インストール モードで起動するために使用するパッケージとプロビジョニング ファイルは、フラッシュに保存する必要があります。usbflash0 または tftp: からインストール モードで起動することはサポートされていません。
- 

### バンドルモードでのブート

バンドル (.bin) ファイルを使用して、device をバンドル モードでブートできます:

```
switch: boot flash:cat9k_iosxe.16.05.01a.SPA.bin
```

バンドルに含まれるプロビジョニングファイルは、どのパッケージを起動、マウント、および実行するかを判断するために使用されます。パッケージはバンドルから取得され、RAM にコピーされます。各パッケージの ISO ファイル システムは、ルート ファイル システムにマウントされます。

インストール モードでの起動とは異なり、バンドル モードでの起動では、バンドルのサイズに対応するサイズの追加メモリが使用されます。

## ブートモードの変更

バンドルブートモードで実行中のデバイスをインストールモードに変更するには、ブート変数を `flash:packages.conf` に設定して **install add file flash:cat9k\_2.bin activate commit** コマンドを実行します。コマンドの実行後、デバイスはインストールブートモードでリブートします。

## ソフトウェアパッケージのインストール

デバイスにソフトウェアパッケージをインストールするには、**install add**、**install activate**、および **install commit** コマンドを特権 EXEC モードで使用します。

**install add** コマンドは、ソフトウェアパッケージをローカルまたはリモートの場所からデバイスにコピーします。FTP、HTTP、HTTPs、または TFTP を使用できます。このコマンドは、`.bin` ファイルの個々のコンポーネントをサブパッケージと `packages.conf` ファイルに抽出します。またファイルを検証して、イメージファイルがプラットフォームに固有であることを確認します。

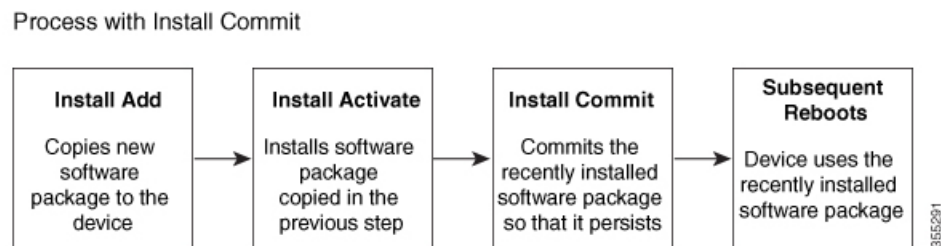
**install activate** コマンドを動作させるには、パッケージをデバイスのブートフラッシュで使用可能にする必要があります。このコマンドを設定すると、`.bin` ファイルから以前に追加したパッケージがアクティブ化され、システムがリロードします。

**install commit** コマンドを有効化して、更新プログラムをリロード全体にわたって確定します。

更新をインストールすると、以前にインストールしたソフトウェアイメージが置換されます。デバイスには常に 1 つのイメージのみがインストールされます。

次のフローチャートで、ソフトウェアのインストールの動作を説明します。

図 1: ソフトウェアパッケージのコミット



(注) **install activate** コマンドは、新しいイメージを使用してデバイスをリロードします。

## ソフトウェアインストールの終了

ソフトウェアイメージのアクティブ化は次の方法で終了できます。

- **install activate auto-abort-timer** コマンドを使用します。新しいイメージをアクティブ化した後にデバイスをリロードすると、`auto-abort-timer` がトリガーされます。**install commit** コマンドを発行する前にタイマーが期限切れになった場合、インストールプロセスが終了し

ます。デバイスは再度リロードし、前のバージョンのソフトウェアイメージで起動します。

このタイマーを停止するには、**install auto-abort-timer stop** コマンドを使用します。

- **install abort** コマンドを使用します。このコマンドは、新しいソフトウェアのインストール前に実行していたバージョンにロールバックします。このコマンドは、**install commit** コマンドを発行する前に使用します。

## Devices 情報の割り当て

IP 情報を割り当てるには、**device** のセットアッププログラムを使用する方法、DHCP サーバーを使用する方法、または手動で実行する方法があります。

特定の IP 情報の設定が必要な場合、**device** のセットアッププログラムを使用してください。このプログラムを使用すると、ホスト名とイネーブルシークレットパスワードを設定することもできます。

また、任意で、Telnet パスワードを割り当てたり（リモート管理中のセキュリティ確保のため）、スイッチをクラスタのコマンドまたはメンバスイッチとして、あるいはスタンドアロンスイッチとして設定したりできます。

サーバーの設定後は DHCP サーバーを使用して、IP 情報の集中管理と自動割り当てを行います。



- (注) DHCP を使用している場合は、**device** が動的に割り当てられた IP アドレスを受信してコンフィギュレーションファイルを読み込むまでは、セットアッププログラムからの質問に回答しないでください。

**device** の設定手順を熟知している経験豊富なユーザーの場合は、**device** を手動で設定してください。それ以外のユーザは、「ブートプロセス」で説明したセットアッププログラムを使用してください。

## デフォルトのスイッチ情報

表 1: デフォルトのスイッチ情報

機能	デフォルト設定
IP アドレスおよびサブネットマスク	IP アドレスまたはサブネット マスクは定義されていません。
デフォルト ゲートウェイ	デフォルト ゲートウェイは定義されていません。
イネーブルシークレットパスワード	パスワードは定義されていません。

機能	デフォルト設定
ホスト名	出荷時に割り当てられるデフォルトのホスト名はデバイスです。
Telnet パスワード	パスワードは定義されていません。
クラスタ コマンド スイッチ機能	ディセーブル
クラスタ名	クラスタ名は定義されません。

## DHCP ベースの自動設定の概要

DHCPは、インターネットホストおよびインターネットワーキングデバイスに設定情報を提供します。このプロトコルには、2つのコンポーネントがあります。1つはDHCPサーバからデバイスにコンフィギュレーションパラメータを提供するコンポーネント、もう1つはデバイスにネットワークアドレスを割り当てるコンポーネントです。DHCPはクライアント/サーバモデルに基づいています。指定されたDHCPサーバが、動的に設定されるデバイスに対して、ネットワークアドレスを割り当て、コンフィギュレーションパラメータを提供します。deviceは、DHCPクライアントおよびDHCPサーバとして機能できます。

DHCPベースの自動設定では、device（DHCPクライアント）は起動時に、IPアドレス情報およびコンフィギュレーションファイルを使用して自動的に設定されます。

DHCPベースの自動設定を使用すると、device上でDHCPクライアント側の設定を行う必要はありません。ただし、DHCPサーバで、IPアドレスに関連した各種リースオプションを設定する必要があります。

DHCPを使用してネットワーク上のコンフィギュレーションファイルの場所をリレーする場合は、TFTPサーバおよびドメインネームシステム（DNS）サーバの設定が必要になることがあります。



- (注) スイッチスタックとDHCP、DNS、TFTPサーバの間では冗長接続を確立することを推奨します。接続されているスタックメンバーがスイッチスタックから削除された場合でも、これらのサーバがアクセス可能なまま維持されるように保証するうえで役立ちます。

deviceのDHCPサーバは、スイッチと同じLAN上に配置することも、そのdeviceとは別のLAN上に配置することもできます。DHCPサーバが異なるLAN上で動作している場合、deviceとDHCPサーバ間に、DHCPのリレーデバイスを設定する必要があります。リレーデバイスは、直接接続されている2つのLAN間でブロードキャストトラフィックを転送しません。ルータはブロードキャストパケットを転送しませんが、受信したパケットの宛先IPアドレスに基づいてパケットを転送します。

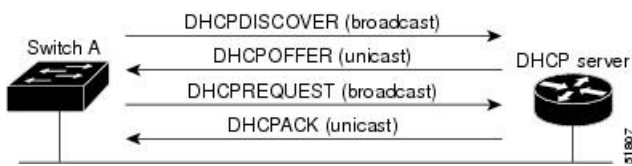
DHCPベースの自動設定は、deviceのBOOTPクライアント機能に代わるものです。

## DHCP クライアントの要求プロセス

deviceを起動したときに、deviceにコンフィギュレーションファイルがない場合、DHCP クライアントが呼び出され、DHCPクライアントがDHCPサーバーに設定情報を要求します。コンフィギュレーションファイルが存在し、その設定に特定のルーテッドインターフェイスの **ip address dhcp** インターフェイス コンフィギュレーション コマンドが含まれる場合、DHCP クライアントが呼び出され、DHCPクライアントがインターフェイスにIPアドレス情報を要求します。

次は、DHCPクライアントとDHCPサーバの間で交換される一連のメッセージです。

図 2: DHCPクライアント/サーバ間のメッセージ交換



クライアントであるデバイスAは、DHCPサーバーの場所を特定するために、DHCPDISCOVERメッセージをブロードキャストします。DHCPサーバは、DHCPOFFERユニキャストメッセージによって、使用可能なコンフィギュレーションパラメータ（IPアドレス、サブネットマスク、ゲートウェイIPアドレス、DNS IPアドレス、IPアドレス用のリースなど）をクライアントに提示します。

DHCPREQUESTブロードキャストメッセージでは、クライアントは、提示された設定情報に対して、DHCPサーバに正式な要求を戻します。この正式な要求はブロードキャストされるため、クライアントからDHCPDISCOVERブロードキャストメッセージを受信した他のすべてのDHCPサーバは、クライアントに提示したIPアドレスを再利用できます。

DHCPサーバは、DHCPACKユニキャストメッセージをクライアントに戻すことで、IPアドレスがクライアントに割り当てられたことを確認します。このメッセージによって、クライアントとサーバはバウンドされ、クライアントはサーバから受信した設定情報を使用します。deviceの受信する情報量は、DHCPサーバの設定方法によって異なります。

DHCPOFFERユニキャストメッセージによって送信されたコンフィギュレーションパラメータが無効である（コンフィギュレーションエラーがある）場合、クライアントはDHCPサーバに、DHCPDECLINEブロードキャストメッセージを戻します。

DHCPサーバはクライアントに、提示されたコンフィギュレーションパラメータが割り当てられていない、パラメータのネゴシエーション中にエラーが発生した、またはDHCPOFFERメッセージに対するクライアントの応答が遅れている（DHCPサーバがパラメータを別のクライアントに割り当てた）という意味のDHCPNAK拒否ブロードキャストメッセージを送信します。

DHCPクライアントは、複数のDHCPサーバまたはBOOTPサーバから提示を受け取り、そのうちの任意の1つを受け入れることができますが、通常は最初に受け取った提示を受け入れません。DHCPサーバから提示されたIPアドレスが必ずしもクライアントに割り当てられるわけではありません。ただし、サーバは通常、クライアントが正式にアドレスを要求するまではアドレスを確保しておきます。deviceがBOOTPサーバからの応答を受け入れ、自身を設定する場合、deviceはdeviceコンフィギュレーションファイルを取得するために、TFTP要求をユニキャストするのではなくブロードキャストします。

DHCP ホスト名オプションにより、`devices`のグループはホスト名および標準コンフィギュレーションを集中管理型 DHCP サーバーから取得できます。クライアント (device) は DHCPDISCOVER メッセージ内に、DHCP サーバーからのホスト名および他のコンフィギュレーションパラメータの要求に使用される **Option 12** フィールドを加えます。すべてのクライアントのコンフィギュレーションファイルは、DHCP から取得したホスト名を除き、まったく同じです。

## DHCP ベースの自動設定およびイメージアップデート

DHCP イメージアップグレード機能を使用すると、ネットワーク内の1つ以上の`devices`に新しいイメージファイルおよび新しいコンフィギュレーションファイルをダウンロードするように DHCP サーバーを設定できます。ネットワーク内のすべてのスイッチでのイメージおよびコンフィギュレーションの同時アップグレードによって、ネットワークに加えられたそれぞれの新しい device が、同じイメージとコンフィギュレーションを確実に受信するようになります。

DHCP イメージアップグレードには、自動設定およびイメージアップデートの2つのタイプがあります。

### DHCP ベースの自動設定の制約事項

- ネットワーク内に割り当てられた IP アドレスがなく、1つ以上のレイヤ3 インターフェイスが起動していない場合は、設定プロセスが保存された DHCP ベースの自動設定は停止します。
- タイムアウトを設定しない限り、設定機能を備えている DHCP ベースの自動設定は IP アドレスのダウンロードを無期限に繰り返します。
- コンフィギュレーションファイルをダウンロードできないか破損している場合は、自動インストールプロセスが停止します。
- TFTP からダウンロードされたコンフィギュレーションファイルは、実行コンフィギュレーション内の既存コンフィギュレーションとマージされますが、**write memory** または **copy running-configuration startup-configuration** 特権 EXEC コマンドを入力しない限り、NVRAM に保存されません。ダウンロードされたコンフィギュレーションがスタートアップコンフィギュレーションに保存された場合、後続のシステム再起動中にこの機能はトリガーされません。

### DHCP 自動設定

DHCP 自動設定は、コンフィギュレーションファイルを DHCP サーバーからネットワーク内の1つ以上の`devices`にダウンロードします。ダウンロードされたコンフィギュレーションファイルは、`device`の実行コンフィギュレーションファイルになります。このファイルは、`device`がリロードされるまで、フラッシュメモリに保存されたブートアップコンフィギュレーションを上書きしません。



## DHCP 自動イメージアップデート

DHCP自動設定とともにDHCP自動イメージアップグレードを使用すると、コンフィギュレーションおよび新しいイメージをネットワーク内の1つ以上のdevicesにダウンロードできます。新しいコンフィギュレーションおよび新しいイメージをダウンロードしている1つのdeviceスイッチ（または複数のdevices）は、ブランク（つまり、出荷時のデフォルト設定がロードされている状態）にできます。

コンフィギュレーションをすでに持っているスイッチに新しいコンフィギュレーションをダウンロードすると、ダウンロードされたコンフィギュレーションは、スイッチに保存されているコンフィギュレーションファイルに追加されます（どの既存のコンフィギュレーションファイルも、ダウンロードされたファイルに上書きされません）。

deviceのDHCP自動イメージアップデートをイネーブルにするには、イメージファイルおよびコンフィギュレーションファイルがあるTFTPサーバーを、正しいオプション67（コンフィギュレーションファイル名）、オプション66（DHCPサーバーホスト名）、オプション150（TFTPサーバーアドレス）、およびオプション125（Cisco IOSイメージファイルの説明）の設定で設定する必要があります。

deviceをネットワークに設置すると、自動イメージアップデート機能が開始します。ダウンロードされたコンフィギュレーションファイルはdeviceの実行コンフィギュレーションに保存され、新しいイメージがダウンロードされてdeviceにインストールされます。deviceを再起動すると、このコンフィギュレーションがdeviceのコンフィギュレーションに保存されます。

## DHCP サーバ設定時の注意事項

デバイスをDHCPサーバとして設定する場合、次の注意事項に従ってください。

- DHCPサーバーには、deviceのハードウェアアドレスによって各deviceと結び付けられている予約済みのリースを設定する必要があります。
- deviceにIPアドレス情報を受信させるには、DHCPサーバーに次のリースオプションを設定する必要があります。
  - クライアントのIPアドレス（必須）
  - クライアントのサブネットマスク（必須）
  - DNSサーバのIPアドレス（任意）
  - ルータのIPアドレス（deviceで使用するデフォルトゲートウェイアドレス）（必須）
- deviceにTFTPサーバーからコンフィギュレーションファイルを受信させる場合は、DHCPサーバーに次のリースオプションを設定する必要があります。
  - TFTPサーバ名（必須）
  - ブートファイル名（クライアントが必要とするコンフィギュレーションファイル名）（推奨）
  - ホスト名（任意）

- DHCP サーバーの設定によっては、**device**は IP アドレス情報またはコンフィギュレーションファイル、あるいはその両方を受信できます。
- 前述のリースオプションを設定しなかった場合、DHCP サーバーは、設定されたパラメータのみを使用してクライアントの要求に応答します。IP アドレスおよびサブネットマスクが応答に含まれていないと、**device**は設定されません。ルータの IP アドレスまたは TFTP サーバー名が見つからなかった場合、**device**は TFTP 要求をユニキャストしないでブロードキャストする場合があります。その他のリースオプションは、使用できなくても自動設定には影響しません。
- **device**は DHCP サーバーとして動作することができます。デフォルトでは、Cisco IOS DHCP サーバーおよび DHCP リレー エージェント機能は **device**上でイネーブルにされていますが、設定されていません。（これらの機能は動作しません）

## TFTP サーバの目的

DHCP サーバーの設定に基づいて、**device**は TFTP サーバーから 1 つまたは複数のコンフィギュレーションファイルをダウンロードしようとします。TFTP サーバーへの IP 接続に必要なすべてのオプションについて **device**に応答するよう DHCP を設定している場合で、なおかつ、TFTP サーバー名、アドレス、およびコンフィギュレーションファイル名を指定して DHCP サーバーを設定している場合、**device**は指定された TFTP サーバーから指定されたコンフィギュレーションファイルをダウンロードしようとします。

コンフィギュレーションファイル名、および TFTP サーバーを指定しなかった場合、またはコンフィギュレーションファイルをダウンロードできなかった場合は、**device**はファイル名と TFTP サーバーアドレスをさまざまに組み合わせてコンフィギュレーションファイルをダウンロードしようとします。ファイルには、特定のコンフィギュレーションファイル名（存在する場合）と次のファイルが指定されています。**network-config**、**cisconet.cfg**、**hostname.config**、または **hostname.cfg** です。この場合、**hostname**は **device**の現在のホスト名です。使用される TFTP サーバーアドレスには、（存在する場合）指定された TFTP サーバーのアドレス、およびブロードキャストアドレス（255.255.255.255）が含まれています。

**device**が正常にコンフィギュレーションファイルをダウンロードするには、TFTP サーバーのベース ディレクトリに 1 つまたは複数のコンフィギュレーションファイルが含まれていなければなりません。含めることのできるファイルは、次のとおりです。

- DHCP 応答で指定されているコンフィギュレーションファイル（実際の **device** コンフィギュレーションファイル）。
- **network-config** または **cisconet.cfg** ファイル（デフォルトのコンフィギュレーションファイル）
- **router-config** または **ciscotr.cfg** ファイル（これらのファイルには、すべての **devices**に共通のコマンドが含まれています。通常、DHCP および TFTP サーバが適切に設定されていれば、これらのファイルはアクセスされません）

DHCP サーバリース データベースに TFTP サーバ名を指定する場合は、DNS サーバのデータベースに TFTP サーバ名と IP アドレスのマッピングを設定することも必要です。

使用する TFTP サーバーが、**device**とは異なる LAN 上にある場合、または**device**がブロードキャスト アドレスを使用してアクセスした場合（前述のすべての必須情報が DHCP サーバーの応答に含まれていない場合に発生）は、リレーを設定して TFTP サーバーに TFTP パケットを転送する必要があります。適切な解決方法は、必要なすべての情報を使用して DHCP サーバを設定することです。

## DNS サーバの目的

DHCP サーバは、DNS サーバを使用して TFTP サーバ名を IP アドレスに変換します。DNS サーバ上で、TFTP サーバ名から IP アドレスへのマッピングを設定する必要があります。TFTP サーバには、**device**のコンフィギュレーションファイルが存在します。

DHCP の応答時に IP アドレスを取得する DHCP サーバのリース データベースに、DNS サーバの IP アドレスを設定できます。リース データベースには、DNS サーバの IP アドレスを 2 つまで入力できます。

DNS サーバは、**device**と同じ LAN 上に配置することも、別の LAN 上に配置することもできます。DNS サーバが別の LAN 上に存在する場合、**device**はルータを介して DNS サーバにアクセスできなければなりません。

## コンフィギュレーションファイルの入手方法

IP アドレスおよびコンフィギュレーションファイル名が DHCP で専用のリースとして取得できるかどうかに応じて、**device**は次の方法で設定情報を入手します。

- IP アドレスおよびコンフィギュレーションファイル名が、**device**用に予約され、DHCP 応答（1 ファイル読み込み方式）で提供されている場合

**device**は DHCP サーバから、IP アドレス、サブネットマスク、TFTP サーバアドレス、およびコンフィギュレーションファイル名を受信します。**device**は、TFTP サーバにユニキャストメッセージを送信し、指定されたコンフィギュレーションファイルをサーバのベース ディレクトリから取得して、ブートアッププロセスを完了します。

- **device**の IP アドレスおよびコンフィギュレーションファイル名が予約されているが、DHCP 応答に TFTP サーバアドレスが含まれていない場合（1 ファイル読み込み方式）。

**device**は DHCP サーバから、IP アドレス、サブネットマスク、およびコンフィギュレーションファイル名を受信します。**device**は、TFTP サーバにブロードキャストメッセージを送信し、指定されたコンフィギュレーションファイルをサーバのベースディレクトリから取得して、ブートアッププロセスを完了します。

- IP アドレスだけが**device**用に予約され、DHCP 応答で提供されており、コンフィギュレーションファイル名は提供されない場合（2 ファイル読み込み方式）

**device**は DHCP サーバから、IP アドレス、サブネットマスク、および TFTP サーバアドレスを受信します。**device**は、TFTP サーバにユニキャストメッセージを送信し、`network-config` または `cisconet.cfg` のデフォルト コンフィギュレーションファイルを取得します（`network-config` ファイルが読み込めない場合、**device**は `cisconet.cfg` ファイルを読み込みます）。

デフォルト コンフィギュレーション ファイルには、**device**のホスト名から IP アドレスへのマッピングが含まれています。**device**は、ファイルの情報をホスト テーブルに書き込み、ホスト名を入手します。ファイルにホスト名がない場合、**device**は DHCP 応答で指定されたホスト名を使用します。DHCP 応答でホスト名が指定されていない場合、**device**はデフォルトの *Switch* ホスト名として使用します。

デフォルトのコンフィギュレーション ファイルまたは DHCP 応答からホスト名を入手した後、**device**はホスト名と同じ名前のコンフィギュレーションファイル (*network-config* または *cisconet.cfg*) のどちらが先に読み込まれたかに応じて、*hostname-config* または *hostname.cf* を TFTP サーバーから読み込みます。*cisconet.cfg* ファイルが読み込まれている場合は、ホストのファイル名は 8 文字に切り捨てられます。

*network-config*、*cisconet.cfg*、またはホスト名と同じ名前のファイルを読み込むことができない場合、**device**は *router-config* ファイルを読み込みます。*router-config* ファイルを読み込むことができない場合、**device**は *ciscortt.cfg* ファイルを読み込みます。



- (注) DHCP 応答から TFTP サーバーを入手できなかった場合、ユニキャスト伝送によるコンフィギュレーション ファイルの読み込みにすべて失敗した場合、または TFTP サーバー名を IP アドレスに変換できない場合には、**device**は TFTP サーバー要求をブロードキャストします。

## 環境変数の制御方法

通常動作**device**では、9600 bps に設定されているコンソール接続のみを通じてブート ロード モードを開始します。電源コードを再接続中に**device**電源コードを取り外し、[Mode] ボタンを押します。システム LED がグリーン の点滅から点灯したままになったら、[Mode] ボタンを放してもかまいません。ブートローダの**device**プロンプトが表示されます。

**device**のブートローダソフトウェアは不揮発性の環境変数をサポートするため、これらの環境変数を使用して、ブートローダまたはシステムで稼働する他のソフトウェアの動作を制御できます。ブートローダの環境変数は、UNIX または DOS システムで設定できる環境変数と類似しています。

値を持つ環境変数は、フラッシュ ファイル システムの外にあるフラッシュ メモリに保存されます。

ファイルの各行には、環境変数名と等号に続いて、その変数の値が指定されます。変数が存在しない場合は、変数の値はありません。値がヌルストリングと表示された場合は、変数に値が設定されています。ヌルストリング (たとえば "") が設定されている変数は、値が設定された変数です。多くの環境変数は事前に定義されており、デフォルト値が設定されています。

環境変数の設定を変更するには、ブートローダにアクセスするか、Cisco IOS コマンドを使用します。通常的环境では、環境変数の設定を変更する必要はありません。

## TFTP の環境変数

イーサネット管理ポートを通してスイッチに PC を接続していると、TFTP でブートローダに対してコンフィギュレーションファイルのアップロードまたはダウンロードができます。このテーブルの環境変数が設定されていることを確認します。

表 2: TFTP の環境変数

変数	説明
MAC_ADDR	<p>スイッチの MAC アドレスを指定します。</p> <p>(注) 変数は変更しないことを推奨します。</p> <p>ただし、ブートローダを稼働した後に変数を変更した場合、またはこの変数が保存されている値と異なる場合は、TFTP を使用する前にこのコマンドを入力します。新しい値を有効にするためにリセットする必要があります。</p>
IP_ADDRESS	<p>スイッチの関連付けられた IP サブネットに IP アドレスおよびサブネットマスクを指定します。</p>
DEFAULT_GATEWAY	<p>デフォルトゲートウェイに IP アドレスおよびサブネットマスクを指定します。</p>

## ソフトウェアイメージのリロードのスケジューリング

device 上でソフトウェアイメージのリロードを後で（深夜、週末など device をあまり使用しないときに）行うように、スケジュールを設定できます。または（ネットワーク内のすべての devices でソフトウェアをアップグレードする場合など）ネットワーク全体でリロードを同時に行うことができます。



(注) リロードのスケジュールは、約 24 日以内に設定する必要があります。

リロードオプションには以下のものがあります。

- 指定した分数、または時間および分数が経過したときに、ソフトウェアがリロードされます。リロードは、約 24 時間以内に実行する必要があります。最大 255 文字で、リロードの理由を指定できます。
- ソフトウェアのリロードが（24時間制で）指定された時間に有効になります。月日を指定すると、指定された日時にリロードが行われるようにスケジュールが設定されます。月日を指定しなかった場合、リロードは当日の指定時刻に行われます（指定時刻が現時刻より後の場合）。または翌日の指定時刻に行われます（指定時刻が現在時刻よりも前の場合）。00:00 を指定すると、深夜 0 時のリロードが設定されます。

**reload** コマンドはシステムを停止させます。手動で起動することが設定されていない限り、システムは自動的に再起動します。

手動で起動するように **device** が設定されている場合、仮想端末からリロードを実行しないでください。これは **device** がブートローダモードになることでリモートユーザーが制御を失う事態を防止するための制約です。

コンフィギュレーションファイルを変更すると、リロードの前にコンフィギュレーションを保存するように指示するプロンプトが **device** により表示されます。保存操作時に、**CONFIG\_FILE** 環境変数がすでに存在しないスタートアップ コンフィギュレーション ファイルを示していた場合、保存を続行するかどうかという問い合わせがシステムから出されます。その状況のまま続けると、リロード時にセットアップ モードが開始されます。

スケジュールがすでに設定されたリロードを取り消すには、**reload cancel** 特権 EXEC コマンドを使用します。

## デバイスセットアップ設定の実行方法

DHCP を使用して **device** に新しいイメージおよび新しいコンフィギュレーションをダウンロードするには、少なくとも 2 つの **devices** を設定する必要があります。1 つ目の **device** は DHCP サーバーおよび TFTP サーバーと同じように機能し、2 つ目の **device** (クライアント) は新しいコンフィギュレーション ファイル、または新しいコンフィギュレーション ファイルおよび新しいイメージ ファイルをダウンロードするように設定されています。

### DHCP 自動設定 (コンフィギュレーション ファイルだけ) の設定

このタスクでは、新しい **device** の自動設定をサポートできるように、ネットワーク内の既存の **device** で TFTP や DHCP の設定の DHCP 自動設定を行う方法を示します。

#### 手順の概要

1. **configure terminal**
2. **ip dhcp pool poolname**
3. **boot filename**
4. **network network-number mask prefix-length**
5. **default-router address**
6. **option 150 address**
7. **exit**
8. **tftp-server flash:filename.text**
9. **interface interface-id**
10. **no switchport**
11. **ip address address mask**
12. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<b>configure terminal</b> 例 :  デバイス# <b>configure terminal</b>	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	<b>ip dhcp pool poolname</b> 例 :  デバイス (config)# <b>ip dhcp pool pool</b>	DHCP サーバアドレス プールの名前を作成し、DHCP プール コンフィギュレーション モードを開始します。
ステップ 3	<b>boot filename</b> 例 :  デバイス (dhcp-config)# <b>boot config-boot.text</b>	ブートイメージとして使用されるコンフィギュレーション ファイルの名前を指定します。
ステップ 4	<b>network network-number mask prefix-length</b> 例 :  デバイス (dhcp-config)# <b>network 10.10.10.0 255.255.255.0</b>	DHCP アドレス プールのサブネット ネットワーク番号およびマスクを指定します。  (注) プレフィックス長は、アドレス プレフィックスを構成するビット数を指定します。プレフィックスは、クライアントのネットワーク マスクを指定する二者択一の方法です。プレフィックス長は、スラッシュ (/) で開始する必要があります。
ステップ 5	<b>default-router address</b> 例 :  デバイス (dhcp-config)# <b>default-router 10.10.10.1</b>	DHCP クライアントのデフォルト ルータの IP アドレスを指定します。
ステップ 6	<b>option 150 address</b> 例 :  デバイス (dhcp-config)# <b>option 150 10.10.10.1</b>	TFTP サーバの IP アドレスを指定します。
ステップ 7	<b>exit</b> 例 :	グローバル コンフィギュレーション モードに戻ります。

	コマンドまたはアクション	目的
	デバイス (dhcp-config) # <b>exit</b>	
ステップ 8	<b>tftp-server flash:filename.text</b> 例：  デバイス (config) # <b>tftp-server flash:config-boot.text</b>	TFTPサーバ上のコンフィギュレーションファイルを指定します。
ステップ 9	<b>interface interface-id</b> 例：  デバイス (config) # <b>interface gigabitethernet1/0/4</b>	コンフィギュレーションファイルを受信するクライアントのアドレスを指定します。
ステップ 10	<b>no switchport</b> 例：  デバイス (config-if) # <b>no switchport</b>	インターフェイスをレイヤ 3 モードにします。
ステップ 11	<b>ip address address mask</b> 例：  デバイス (config-if) # <b>ip address 10.10.10.1 255.255.255.0</b>	IP アドレスとインターフェイスのマスクを指定します。
ステップ 12	<b>end</b> 例：  デバイス (config-if) # <b>end</b>	特権 EXEC モードに戻ります。

## DHCP 自動イメージアップデート（コンフィギュレーションファイルおよびイメージ）の設定

このタスクでは、新しいスイッチのインストールをサポートするように既存の device で TFTP および DHCP を設定する DHCP 自動設定について説明します。



### 始める前に

最初にdevice1にアップロードするテキスト ファイル（たとえば、autoinstall\_dhcp）を作成します。このテキストファイル内に、ダウンロードするイメージの名前を含めます（たとえば、cat9k\_iosxe.16.xx.xx.SPA.bin）。

### 手順の概要

1. **configure terminal**
2. **ip dhcp pool poolname**
3. **boot filename**
4. **network network-number mask prefix-length**
5. **default-router address**
6. **option 150 address**
7. **option 125 hex**
8. **copy tftp flash filename.txt**
9. **copy tftp flash imagename.bin**
10. **exit**
11. **tftp-server flash: config.txt**
12. **tftp-server flash: imagename.bin**
13. **tftp-server flash: filename.txt**
14. **interface interface-id**
15. **no switchport**
16. **ip address address mask**
17. **end**
18. **copy running-config startup-config**

### 手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<b>configure terminal</b> 例：  デバイス# <b>configure terminal</b>	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	<b>ip dhcp pool poolname</b> 例：  デバイス (config)# <b>ip dhcp pool pool1</b>	DHCP サーバアドレス プールの名前を作成し、DHCP プール コンフィギュレーション モードを開始します。
ステップ 3	<b>boot filename</b> 例：  デバイス (dhcp-config)# <b>boot config-boot.text</b>	ブート イメージとして使用されるファイルの名前を指定します。

	コマンドまたはアクション	目的
ステップ 4	<p><b>network</b> <i>network-number mask prefix-length</i></p> <p>例 :</p> <p>デバイス (dhcp-config) # <b>network 10.10.10.0 255.255.255.0</b></p>	<p>DHCP アドレス プールのサブネット ネットワーク番号およびマスクを指定します。</p> <p>(注)     プレフィックス長は、アドレスプレフィックスを構成するビット数を指定します。プレフィックスは、クライアントのネットワーク マスクを指定する二者択一の方法です。プレフィックス長は、スラッシュ (/) で開始する必要があります。</p>
ステップ 5	<p><b>default-router</b> <i>address</i></p> <p>例 :</p> <p>デバイス (dhcp-config) # <b>default-router 10.10.10.1</b></p>	<p>DHCP クライアントのデフォルト ルータの IP アドレスを指定します。</p>
ステップ 6	<p><b>option 150</b> <i>address</i></p> <p>例 :</p> <p>デバイス (dhcp-config) # <b>option 150 10.10.10.1</b></p>	<p>TFTP サーバの IP アドレスを指定します。</p>
ステップ 7	<p><b>option 125</b> <i>hex</i></p> <p>例 :</p> <p>デバイス (dhcp-config) # <b>option 125 hex 0000.0009.0a05.08661.7574.6f69.6e73.7461.6c6c.5f64.686370</b></p>	<p>イメージ ファイルのパスを記述したテキスト ファイルのパスを指定します。</p>
ステップ 8	<p><b>copy tftp flash</b> <i>filename.txt</i></p> <p>例 :</p> <p>デバイス (config) # <b>copy tftp flash image.bin</b></p>	<p>device に、テキスト ファイルをアップロードします。</p>
ステップ 9	<p><b>copy tftp flash</b> <i>imagename.bin</i></p> <p>例 :</p> <p>デバイス (config) # <b>copy tftp flash image.bin</b></p>	<p>device に、新しいイメージの tar ファイルをアップロードします。</p>
ステップ 10	<p><b>exit</b></p> <p>例 :</p>	<p>グローバル コンフィギュレーション モードに戻ります。</p>

	コマンドまたはアクション	目的
	デバイス (dhcp-config) # <b>exit</b>	
ステップ 11	<b>tftp-server flash: config.text</b> 例 :  デバイス (config) # <b>tftp-server flash:config-boot.text</b>	TFTP サーバ上の Cisco IOS コンフィギュレーション ファイルを指定します。
ステップ 12	<b>tftp-server flash: imagename.bin</b> 例 :  デバイス (config) # <b>tftp-server flash:image.bin</b>	TFTP サーバ上のイメージ名を指定します。
ステップ 13	<b>tftp-server flash: filename.txt</b> 例 :  デバイス (config) # <b>tftp-server flash:boot-config.text</b>	ダウンロードするイメージファイルの名前を記述したテキスト ファイルを指定します。
ステップ 14	<b>interface interface-id</b> 例 :  デバイス (config) # <b>interface gigabitEthernet1/0/4</b>	コンフィギュレーション ファイルを受信するクライアントのアドレスを指定します。
ステップ 15	<b>no switchport</b> 例 :  デバイス (config-if) # <b>no switchport</b>	インターフェイスをレイヤ 3 モードにします。
ステップ 16	<b>ip address address mask</b> 例 :  デバイス (config-if) # <b>ip address 10.10.10.1 255.255.255.0</b>	IP アドレスとインターフェイスのマスクを指定します。
ステップ 17	<b>end</b> 例 :  デバイス (config-if) # <b>end</b>	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
ステップ 18	<b>copy running-config startup-config</b> 例 :  デバイス (config-if) # <b>end</b>	(任意) コンフィギュレーション ファイルに設定を保存します。

## DHCP サーバからファイルをダウンロードするクライアントの設定



(注) レイヤ3インターフェイスだけを設定してイネーブルにする必要があります。保存されているコンフィギュレーションのDHCPベースの自動設定にIPアドレスを割り当てないでください。

### 手順の概要

1. **configure terminal**
2. **boot host dhcp**
3. **boot host retry timeout *timeout-value***
4. **banner config-save ^C *warning-message* ^C**
5. **end**
6. **show boot**

### 手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<b>configure terminal</b> 例 :  デバイス # <b>configure terminal</b>	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	<b>boot host dhcp</b> 例 :  デバイス (conf) # <b>boot host dhcp</b>	保存されているコンフィギュレーションで自動設定をイネーブルにします。
ステップ 3	<b>boot host retry timeout <i>timeout-value</i></b> 例 :  デバイス (conf) # <b>boot host retry timeout 300</b>	(任意) システムがコンフィギュレーション ファイルをダウンロードしようとする時間を設定します。  (注) タイムアウトを設定しないと、システムは無期限に DHCP サーバから IP アドレスを取得しようとします。

	コマンドまたはアクション	目的
ステップ 4	<b>banner config-save ^C warning-message ^C</b> 例 :  デバイス (conf) # <b>banner config-save ^C Caution - Saving Configuration File to NVRAM May Cause You to No longer Automatically Download Configuration Files at Reboot^C</b>	(任意) コンフィギュレーション ファイルを NVRAM に保存しようとするときに表示される警告メッセージを作成します。
ステップ 5	<b>end</b> 例 :  デバイス (config-if) # <b>end</b>	特権 EXEC モードに戻ります。
ステップ 6	<b>show boot</b> 例 :  デバイス # <b>show boot</b>	設定を確認します。

## 複数の SVI への IP 情報の手動割り当て

このタスクでは、複数のスイッチ仮想インターフェイス (SVI) に IP 情報を手動で割り当てる方法について説明します。

### 手順の概要

1. **configure terminal**
2. **interface vlan *vlan-id***
3. **ip address *ip-address subnet-mask***
4. **exit**
5. **ip default-gateway *ip-address***
6. **end**
7. **show interfaces vlan *vlan-id***
8. **show ip redirects**

### 手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<b>configure terminal</b> 例 :  デバイス # <b>configure terminal</b>	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。

	コマンドまたはアクション	目的
ステップ 2	<b>interface vlan <i>vlan-id</i></b> 例 :  デバイス (config) # <b>interface vlan 99</b>	インターフェイス コンフィギュレーション モードを開始して、IP 情報が割り当てられている VLAN を指定します。指定できる範囲は 1 ~ 4094 です。
ステップ 3	<b>ip address <i>ip-address subnet-mask</i></b> 例 :  デバイス (config-vlan) # <b>ip address 10.10.10.2 255.255.255.0</b>	IP アドレスとサブネット マスクを入力します。
ステップ 4	<b>exit</b> 例 :  デバイス (config-vlan) # <b>exit</b>	グローバル コンフィギュレーション モードに戻ります。
ステップ 5	<b>ip default-gateway <i>ip-address</i></b> 例 :  デバイス (config) # <b>ip default-gateway 10.10.10.1</b>	device に直接接続しているネクスト ホップのルータ インターフェイスの IP アドレスを入力します。このスイッチにはデフォルトゲートウェイが設定されています。デフォルトゲートウェイは、device スイッチから宛先 IP アドレスを取得していない IP パケットを受信します。  デフォルトゲートウェイが設定されると、device は、ホストが接続する必要のあるリモートネットワークに接続できます。  (注) IP でルーティングするように device を設定した場合、デフォルトゲートウェイの設定は不要です。  (注) デフォルトゲートウェイの構成に基づいて、device の CAPWAP は中継を行い、ルーティングされたアクセスポイントと device の接続をサポートします。
ステップ 6	<b>end</b> 例 :  デバイス (config) # <b>end</b>	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
ステップ 7	<b>show interfaces vlan <i>vlan-id</i></b> 例： デバイス# <code>show interfaces vlan 99</code>	設定された IP アドレスを確認します。
ステップ 8	<b>show ip redirects</b> 例： デバイス# <code>show ip redirects</code>	設定されたデフォルトゲートウェイを確認します。

## デバイスのスタートアップコンフィギュレーションの変更

### システムコンフィギュレーションを読み書きするためのファイル名の指定

Cisco IOS ソフトウェアは、デフォルトで `config.text` ファイルを使用して、システムコンフィギュレーションの不揮発性コピーを読み書きします。別のファイル名を指定することもできます。次回の起動時には、その名前のファイルが読み込まれます。

#### 始める前に

このタスクではスタンドアロンの device を使用します。

#### 手順の概要

1. **configure terminal**
2. **boot flash:/file-url**
3. **end**
4. **show boot**
5. **copy running-config startup-config**

#### 手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<b>configure terminal</b> 例： Switch# <code>configure terminal</code>	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	<b>boot flash:/file-url</b> 例：	次回の起動時に読み込むコンフィギュレーションファイルを指定します。

スイッチの手動による起動

	コマンドまたはアクション	目的
	Switch(config)# <b>boot flash:config.text</b>	<i>file-url</i> : パス (ディレクトリ) およびコンフィギュレーションファイル名。 ファイル名およびディレクトリ名は、大文字と小文字を区別します。
ステップ 3	<b>end</b> 例 : Switch(config)# <b>end</b>	特権 EXEC モードに戻ります。
ステップ 4	<b>show boot</b> 例 : Switch# <b>show boot</b>	入力を確認します。 <b>boot</b> グローバル コンフィギュレーション コマンドによって、CONFIG_FILE 環境変数の設定が変更されます。
ステップ 5	<b>copy running-config startup-config</b> 例 : Switch# <b>copy running-config startup-config</b>	(任意) コンフィギュレーションファイルに設定を保存します。

## スイッチの手動による起動

スイッチはデフォルトで自動的に起動しますが、手動で起動するように設定することもできます。

始める前に

このタスクのスタンドアロンスイッチを使用します。

### 手順の概要

1. **configure terminal**
2. **boot manual**
3. **end**
4. **show boot**
5. **copy running-config startup-config**

### 手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<b>configure terminal</b> 例 :	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。



	コマンドまたはアクション	目的
	デバイス# <code>configure terminal</code>	
ステップ 2	<p><b>boot manual</b></p> <p>例 :</p> <p>デバイス(config)# <code>boot manual</code></p>	<p>次の起動時に、スイッチを手動で起動できるようにします。</p>
ステップ 3	<p><b>end</b></p> <p>例 :</p> <p>デバイス(config)# <code>end</code></p>	<p>特権 EXEC モードに戻ります。</p>
ステップ 4	<p><b>show boot</b></p> <p>例 :</p> <p>デバイス# <code>show boot</code></p>	<p>入力を確認します。</p> <p><b>boot manual</b> グローバルコマンドは、<b>MANUAL_BOOT</b> 環境変数の設定を変更します。</p> <p>次回、システムを再起動した際には、スイッチはブートローダモードになり、ブートローダモードであることが <i>switch:</i> プロンプトによって示されます。システムを起動するには、<b>boot filesystem:/file-url</b> ブートローダコマンドを使用します。</p> <ul style="list-style-type: none"> <li>• <i>filesystem</i> : システムボードのフラッシュデバイスに <i>flash:</i> を使用します。</li> </ul> <p>Switch: <b>boot flash:</b></p> <ul style="list-style-type: none"> <li>• <i>file-url</i> : パス（ディレクトリ）および起動可能なイメージの名前を指定します。</li> </ul> <p>ファイル名およびディレクトリ名は、大文字と小文字を区別します。</p>
ステップ 5	<p><b>copy running-config startup-config</b></p> <p>例 :</p> <p>デバイス# <code>copy running-config startup-config</code></p>	<p>(任意) コンフィギュレーションファイルに設定を保存します。</p>

## インストールモードでのデバイスのブート

### ソフトウェアパッケージのインストール

単一のコマンドまたは個別のコマンドを使用してソフトウェアパッケージをインストールして、アクティブ化し、コミットできます。このタスクでは、ソフトウェアパッケージをインストールするための **install add file activate commit** コマンドの使用方法を示します。

#### 手順の概要

1. **enable**
2. **install add file tftp: filename [activate commit]**
3. **exit**

#### 手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<b>enable</b> 例： Device> enable	特権 EXEC モードを有効にします。  • パスワードを入力します（要求された場合）。
ステップ 2	<b>install add file tftp: filename [activate commit]</b> 例： Device# install add file tftp://172.16.0.1/tftpboot/folder1/ cat9k_iosxe.16.06.01.SPA.bin activate commit	ソフトウェアインストールパッケージをリモートロケーションから（FTP、HTTP、HTTPS、TFTPを介して）デバイスにコピーし、プラットフォームおよびイメージバージョンの互換性チェックを実行し、ソフトウェアパッケージをアクティブ化し、そのパッケージを複数回リロードしても維持されるようにします。  • このコマンドは、.bin ファイルの個別のコンポーネントをサブパッケージと packages.conf ファイルに抽出します。  • このコマンドの実行後にデバイスはリロードします。
ステップ 3	<b>exit</b> 例： Device# exit	特権 EXEC モードを終了し、ユーザー EXEC モードに戻ります。

### 更新プログラムパッケージの管理

#### 手順の概要

1. **enable**
2. **install add file tftp: filename**
3. **install activate [auto-abort-timer]**

4. **install abort**
5. **install commit**
6. **install rollback to committed**
7. **install remove {file filesystem: filename | inactive}**
8. **show install summary**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<b>enable</b> 例： Device> enable	特権 EXEC モードを有効にします。 <ul style="list-style-type: none"> <li>• パスワードを入力します（要求された場合）。</li> </ul>
ステップ 2	<b>install add file tftp: filename</b> 例： Device# install add file tftp://172.16.0.1//tftpboot/folder1/ cat9k_iosxe.16.06.01.SPA.bin	リモートロケーションから（FTP、HTTP、HTTPs、TFTP を介して）デバイスにソフトウェアインストールパッケージをコピーし、プラットフォームとイメージのバージョンの互換性チェックを実行します。 <ul style="list-style-type: none"> <li>• このコマンドは、.bin ファイルの個別のコンポーネントをサブパッケージと packages.conf ファイルに抽出します。</li> </ul>
ステップ 3	<b>install activate [auto-abort-timer]</b> 例： Device# install activate	追加のソフトウェア インストール パッケージをアクティブ化し、デバイスをリロードします。 <ul style="list-style-type: none"> <li>• ソフトウェアの完全インストールを実行する場合は、パッケージファイル名を指定しないでください。</li> <li>• <b>auto-abort-timer</b> キーワードがソフトウェアイメージのアクティブ化を自動的にロールバックします。</li> </ul> 新しいイメージがアクティブになった後で自動タイマーがトリガーされます。 <b>install commit</b> コマンドを発行する前にタイマーの期限が切れた場合、インストールプロセスは自動的に終了します。デバイスがリロードし、以前のバージョンのソフトウェアイメージで起動します。
ステップ 4	<b>install abort</b> 例： Device# install abort	（任意）ソフトウェアインストールのアクティブ化を終了し、現在のインストール手順の前に実行していたバージョンにロールバックします。

デバイスをバンドルモードで起動する場合

	コマンドまたはアクション	目的
		<ul style="list-style-type: none"> <li>このコマンドは、イメージがアクティブ化されている状態でのみ使用できます。イメージがコミットされた状態の場合は使用できません。</li> </ul>
ステップ 5	<b>install commit</b> 例： Device# install commit	リロードが繰り返されても持続する変更を行います。  <ul style="list-style-type: none"> <li><b>install commit</b> コマンドで、新しいイメージのインストールを完了します。自動アポートタイマーが期限切れになるまで、複数回のリロード後も変更は維持されます。</li> </ul>
ステップ 6	<b>install rollback to committed</b> 例： Device# install rollback to committed	(任意) 最後にコミットしたバージョンに更新をロールバックします。
ステップ 7	<b>install remove {file filesystem: filename   inactive}</b> 例： Device# install remove inactive	(任意) 未使用および非アクティブ状態のソフトウェア インストール ファイルを削除します。
ステップ 8	<b>show install summary</b> 例： Device# show install summary	アクティブパッケージに関する情報を表示します。  <ul style="list-style-type: none"> <li>このコマンドの出力は、設定されている <b>install</b> コマンドに応じて変化します。</li> </ul>

## デバイスをバンドルモードで起動する場合

deviceを起動するには、いくつかの方法があります。1つは、TFTP サーバから bin ファイルをコピーしてdeviceを起動する方法です。または、**boot flash:<image.bin>** コマンドか、**boot usbflash0:<image.bin>** コマンドを使用して、deviceをフラッシュまたはUSBフラッシュから直接起動することもできます。

以下の手順は、バンドルモードで TFTP サーバから device を起動する方法を示します。

### 手順の概要

1. **switch:BOOT=<source path of .bin file>**
2. **boot**
3. **show version**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<b>switch:BOOT=&lt;source path of .bin file&gt;</b> 例： switch:BOOT=tftp://10.0.0.2/cat9k_iosxe.16.05.01a.SPA.bin	ブートパラメータを設定します。
ステップ 2	<b>boot</b> 例： switch: boot	deviceをブートします。
ステップ 3	<b>show version</b>	deviceがバンドルモードであることを確認します。

## ソフトウェアイメージのリロードのスケジュール設定

このタスクでは、ソフトウェアイメージを後でリロードするようにdeviceを設定する方法について説明します。

手順の概要

1. **configure terminal**
2. **copy running-config startup-config**
3. **reload in [hh:]mm [text]**
4. **reload at hh: mm [month day | day month] [text]**
5. **reload cancel**
6. **show reload**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<b>configure terminal</b> 例： デバイス# <b>configure terminal</b>	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	<b>copy running-config startup-config</b> 例： デバイス# <b>copy running-config startup-config</b>	<b>reload</b> コマンドを使用する前に、device の設定情報をスタートアップコンフィギュレーションに保存します。
ステップ 3	<b>reload in [hh:]mm [text]</b> 例： デバイス# <b>reload in 12</b> System configuration has been modified. Save?	指定した分数、または時間および分数が経過したときに、ソフトウェアがリロードされるようにスケジュールを設定します。リロードは、約 24 日以内に行う必要があります。最大 255 文字で、リロードの理由を指定できます。

	コマンドまたはアクション	目的
	[yes/no]: <b>y</b>	
ステップ 4	<b>reload at hh: mm [month day   day month] [text]</b> 例： デバイス(config)# <b>reload at 14:00</b>	リロードを実行する時間を、時間数と分数で指定します。 (注) <b>at</b> キーワードを使用するのは、 <b>device</b> システムクロックが（ネットワークタイムプロトコル、ハードウェアカレンダー、または手動で）設定されている場合だけです。時刻は、 <b>device</b> に設定されたタイムゾーンに基づきます。リロードが複数の <b>devices</b> で同時に行われるようにスケジューリングするには、各 <b>device</b> の時間がNTPと同期している必要があります。
ステップ 5	<b>reload cancel</b> 例： デバイス(config)# <b>reload cancel</b>	以前にスケジューリングされたリロードをキャンセルします。
ステップ 6	<b>show reload</b> 例： <b>show reload</b>	以前 <b>device</b> にスケジューリングされたリロードに関する情報、またはリロードがスケジューリングされているかを表示します。

## デバイスのセットアップ設定のモニターリング

### 例: インストールモードでのソフトウェアブートアップディスプレイ

この例では、インストールモードでのソフトウェアブートアップの表示を示します。

```
switch: boot flash:packages.conf
Attempting to boot from [flash:packages.conf]
Located packages.conf
#

validate_package: SHA-1 hash:
    expected 340D5091:2872A0DD:03E9068C:3FDBECAB:69786462
    calculated 340D5091:2872A0DD:03E9068C:3FDBECAB:69786462
Image parsed from conf file is cat9k-rpboot.16.05.01a.SPA.pkg
#####

Waiting for 120 seconds for other switches to boot
#####
Switch number is 1

Restricted Rights Legend
```

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c) of the Commercial Computer Software - Restricted Rights clause at FAR sec. 52.227-19 and subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS sec. 252.227-7013.

cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, California 95134-1706

Cisco IOS Software [Everest], Catalyst L3 Switch Software (CAT9K\_IOSXE), Version 16.5.1a, RELEASE SOFTWARE (fc2)  
Technical Support: <http://www.cisco.com/techsupport>  
Copyright (c) 1986-2017 by Cisco Systems, Inc.  
Compiled Tue 30-May-17 00:36 by mcpre

Cisco IOS-XE software, Copyright (c) 2005-2017 by cisco Systems, Inc. All rights reserved. Certain components of Cisco IOS-XE software are licensed under the GNU General Public License ("GPL") Version 2.0. The software code licensed under GPL Version 2.0 is free software that comes with ABSOLUTELY NO WARRANTY. You can redistribute and/or modify such GPL code under the terms of GPL Version 2.0. For more details, see the documentation or "License Notice" file accompanying the IOS-XE software, or the applicable URL provided on the flyer accompanying the IOS-XE software.

FIPS: Flash Key Check : Begin  
FIPS: Flash Key Check : End, Not Found, FIPS Mode Not Enabled

This product contains cryptographic features and is subject to United States and local country laws governing import, export, transfer and use. Delivery of Cisco cryptographic products does not imply third-party authority to import, export, distribute or use encryption. Importers, exporters, distributors and users are responsible for compliance with U.S. and local country laws. By using this product you agree to comply with applicable laws and regulations. If you are unable to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at: <http://www.cisco.com/wwl/export/crypto/tool/stqrg.html>

If you require further assistance please contact us by sending email to [export@cisco.com](mailto:export@cisco.com).

cisco C9300-48P (X86) processor with 818597K/6147K bytes of memory.  
Processor board ID FCW2049G03S  
2048K bytes of non-volatile configuration memory.  
8388608K bytes of physical memory.  
1638400K bytes of Crash Files at crashinfo:.  
11264000K bytes of Flash at flash:.  
0K bytes of WebUI ODM Files at webui:.

Base Ethernet MAC Address : 04:6c:9d:01:3b:80  
Motherboard Assembly Number : 73-17956-04  
Motherboard Serial Number : FOC20465ABU

## 例: インストールモードでのソフトウェアブートアップディスプレイ

```

Model Revision Number      : P4B
Motherboard Revision Number : 04
Model Number               : C9300-48P
System Serial Number       : FCW2049G03S

```

```
%INIT: waited 0 seconds for NVRAM to be available
```

```
Defaulting CPP : Policer rate for all classes will be set to their defaults
```

```
Press RETURN to get started!
```

この例では、バンドルモードでのソフトウェアブートアップの表示を示します。

```
switch: boot flash:cat9k_iosxe.16.05.01a.SPA.bin
```

```
Attempting to boot from [flash:cat9k_iosxe.16.05.01a.SPA.bin]
```

```
Located cat9k_iosxe.16.05.01a.SPA.bin
```

```
#####
Warning: ignoring ROMMON var "BOOT_PARAM"
```

```
Waiting for 120 seconds for other switches to boot
```

```
#####
Switch number is 3
```

#### Restricted Rights Legend

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c) of the Commercial Computer Software - Restricted Rights clause at FAR sec. 52.227-19 and subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS sec. 252.227-7013.

```

cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134-1706

```

```

Cisco IOS Software [Everest], Catalyst L3 Switch Software (CAT9K_IOSXE), Version 16.5.1a,
RELEASE SOFTWARE (fc2)

```

```
Technical Support: http://www.cisco.com/techsupport
```

```
Copyright (c) 1986-2017 by Cisco Systems, Inc.
```

```
Compiled Tue 30-May-17 00:36 by mcpre
```

Cisco IOS-XE software, Copyright (c) 2005-2017 by Cisco Systems, Inc. All rights reserved. Certain components of Cisco IOS-XE software are licensed under the GNU General Public License ("GPL") Version 2.0. The software code licensed under GPL Version 2.0 is free software that comes with ABSOLUTELY NO WARRANTY. You can redistribute and/or modify such GPL code under the terms of GPL Version 2.0. For more details, see the documentation or "License Notice" file accompanying the IOS-XE software, or the applicable URL provided on the flyer accompanying the IOS-XE software.

```
FIPS: Flash Key Check : Begin
```



```
FIPS: Flash Key Check : End, Not Found, FIPS Mode Not Enabled

This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@cisco.com.

cisco C9300-24U (X86) processor with 818597K/6147K bytes of memory.
Processor board ID FCW2111G00X
2048K bytes of non-volatile configuration memory.
8388608K bytes of physical memory.
1638400K bytes of Crash Files at crashinfo:.
11264000K bytes of Flash at flash:.
15633392K bytes of USB Flash at usbflash0:.
0K bytes of WebUI ODM Files at webui:.

Base Ethernet MAC Address      : 04:6c:9d:1e:2a:80
Motherboard Assembly Number    : 73-17954-05
Motherboard Serial Number      : FOC21094MWL
Model Revision Number          : PP
Motherboard Revision Number    : 05
Model Number                   : C9300-24U
System Serial Number           : FCW2111G00X

%INIT: waited 0 seconds for NVRAM to be available

Defaulting CPP : Policer rate for all classes will be set to their defaults

Press RETURN to get started!
```

## 例：緊急インストール

以下に、**emergency-install** コマンドが開始された場合の出力サンプルの例を示します。

```
switch: emergency-install
tftp://223.255.254.254/auto/tftpboot/X86/cat9k_iosxe.16.05.01a.SPA.bin
WARNING: The system partition (bootflash:) will be erased during the system recovery
install process.
Are you sure you want to proceed? [y] y/n [n]: y
Starting system recovery
(tftp://223.255.254.254/auto/tftpboot/X86/cat9k_iosxe.16.05.01a.SPA.bin) ...
Attempting to boot from [sda9:cat9k-recovery.SSA.bin]
Located cat9k-recovery.SSA.bin
#####
Warning: ignoring ROMMON var "BOOT_PARAM"
```

```

PLATFORM_TYPE C9300 speed 9600

Booting Recovery Image 16.5.1a

Initiating Emergency Installation of bundle
tftp://223.255.254.254/auto/tftpboot/X86/cat9k_iosxe.16.05.01a.SPA.bin

Downloading bundle
tftp://223.255.254.254/auto/tftpboot/X86/cat9k_iosxe.16.05.01a.SPA.bin...
curl_vrf=2
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
 100  485M  100  485M    0     0  5143k      0  0:01:36  0:01:36  --:--:-- 5256k
 100  485M  100  485M    0     0  5143k      0  0:01:36  0:01:36  --:--:-- 5143k

Validating bundle tftp://223.255.254.254/auto/tftpboot/X86/cat9k_iosxe.16.05.01a.SPA.bin...
Installing bundle
tftp://223.255.254.254/auto/tftpboot/X86/cat9k_iosxe.16.05.01a.SPA.bin...
Verifying bundle tftp://223.255.254.254/auto/tftpboot/X86/cat9k_iosxe.16.05.01a.SPA.bin...
Package cat9k-cc_srdriver.16.05.01a.SPA.pkg
/temp//stage/cat9k-cc_srdriver.16.05.01a.SPA.pkg is Digitally Signed
Package cat9k-espbase.16.05.01a.SPA.pkg /temp//stage/cat9k-espbase.16.05.01a.SPA.pkg is
Digitally Signed
Package cat9k-guestshell.16.05.01a.SPA.pkg /temp//stage/cat9k-guestshell.16.05.01a.SPA.pkg
is Digitally Signed
Package cat9k-rpbase.16.05.01a.SPA.pkg /temp//stage/cat9k-rpbase.16.05.01a.SPA.pkg is
Digitally Signed
Package cat9k-sipbase.16.05.01a.SPA.pkg /temp//stage/cat9k-sipbase.16.05.01a.SPA.pkg is
Digitally Signed
Package cat9k-sipspa.16.05.01a.SPA.pkg /temp//stage/cat9k-sipspa.16.05.01a.SPA.pkg is
Digitally Signed
Package cat9k-srdriver.16.05.01a.SPA.pkg /temp//stage/cat9k-srdriver.16.05.01a.SPA.pkg
is Digitally Signed
Package cat9k-webui.16.05.01a.SPA.pkg /temp//stage/cat9k-webui.16.05.01a.SPA.pkg is
Digitally Signed
Package cat9k-wlc.16.05.01a.SPA.pkg /temp//stage/cat9k-wlc.16.05.01a.SPA.pkg is Digitally
Signed
Package /cat9k-rpboot.16.05.01a.SPA.pkg /temp//rpboot/cat9k-rpboot.16.05.01a.SPA.pkg is
Digitally Signed
Preparing flash...
Flash filesystem unmounted successfully /dev/sdb3
Syncing device...
Emergency Install successful... Rebooting
Will reboot now

Initializing Hardware...

System Bootstrap, Version 16.5.2r, RELEASE SOFTWARE (P)
Compiled Wed 05/31/2017 15:58:35.22 by rel

Current image running:
Primary Rommon Image

Last reset cause: SoftwareReload
C9300-24U platform with 8388608 Kbytes of main memory

```

# デバイスのセットアップを実行する場合の設定例

## 例：更新プログラムパッケージの管理

次に、ソフトウェア パッケージ ファイルを追加する例を示します。

```
Device# install add file flash:cat9k_iosxe.16.06.02.SPA.bin activate commit

install_add_activate_commit: START Mon Oct 30 19:54:51 UTC 2017

System configuration has been modified.
Press Yes(y) to save the configuration and proceed.
Press No(n) for proceeding without saving the configuration.
Press Quit(q) to exit, you may save configuration and re-enter the command. [y/n/q]y
Building configuration...

[OK]Modified configuration has been saved

*Oct 30 19:54:55.633: %IOSXE-5-PLATFORM: Switch 1 R0/0: Oct 30 19:54:55 install_engine.sh:

%INSTALL-5-INSTALL_START_INFO: Started install one-shot
flash:cat9k_iosxe.16.06.02.SPA.bininstall_add_activate_commit: Adding PACKAGE

This operation requires a reload of the system. Do you want to proceed?
Please confirm you have changed boot config to flash:packages.conf [y/n]y

--- Starting initial file syncing ---
Info: Finished copying flash:cat9k_iosxe.16.06.02.SPA.bin to the selected switch(es)
Finished initial file syncing

--- Starting Add ---
Performing Add on all members
  [1] Add package(s) on switch 1
  [1] Finished Add on switch 1
Checking status of Add on [1]
Add: Passed on [1]
Finished Add

install_add_activate_commit: Activating PACKAGE
Following packages shall be activated:
/flash/cat9k-wlc.16.06.02.SPA.pkg
/flash/cat9k-webui.16.06.02.SPA.pkg
/flash/cat9k-srdriver.16.06.02.SPA.pkg
/flash/cat9k-sipspa.16.06.02.SPA.pkg
/flash/cat9k-sipbase.16.06.02.SPA.pkg
/flash/cat9k-rpboot.16.06.02.SPA.pkg
/flash/cat9k-rpbase.16.06.02.SPA.pkg
/flash/cat9k-guestshell.16.06.02.SPA.pkg
/flash/cat9k-espbase.16.06.02.SPA.pkg
/flash/cat9k-cc_srdriver.16.06.02.SPA.pkg

This operation requires a reload of the system. Do you want to proceed? [y/n]y
--- Starting Activate ---
Performing Activate on all members
  [1] Activate package(s) on switch 1
  [1] Finished Activate on switch 1
Checking status of Activate on [1]
Activate: Passed on [1]
Finished Activate
```

```

--- Starting Commit ---
Performing Commit on all members

*Oct 30 19:57:41.145: %IOSXE-5-PLATFORM: Switch 1 R0/0: Oct 30 19:57:41 rollback_timer.sh:

%INSTALL-5-INSTALL_AUTO_ABORT_TIMER_PROGRESS: Install auto abort timer will expire in
7200 seconds [1]
Commit package(s) on switch 1
  [1] Finished Commit on switch 1
Checking status of Commit on [1]
Commit: Passed on [1]
Finished Commit

Install will reload the system now!
SUCCESS: install_add_activate_commit Mon Oct 30 19:57:48 UTC 2017

Device#
*Oct 30 19:57:48.384: %IOSXE-5-PLATFORM: Switch 1 R0/0: Oct 30 19:57:48 install_engine.sh:

%INSTALL-5-INSTALL_COMPLETED_INFO: Completed install one-shot PACKAGE
flash:cat9k_iosxe.16.06.02.SPA.bin

Chassis 1 reloading, reason - Reload command

```

次に、ソフトウェアパッケージファイルをデバイスに追加した後の **show install summary** コマンドの出力例を示します。

```

Device# show install summary

[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
             C - Activated & Committed, D - Deactivated & Uncommitted
-----
Type  St   Filename/Version
-----
IMG   I    16.6.1.0
IMG   C    16.6.2.0

```

次に、追加したソフトウェアパッケージファイルをアクティブ化する例を示します。

```

Device# install activate

install_activate: START Mon Oct 30 20:14:20 UTC 2017
install_activate: Activating PACKAGE

*Oct 30 20:14:21.379: %IOSXE-5-PLATFORM: Switch 1 R0/0: Oct 30 20:14:21 install_engine.sh:

%INSTALL-5-INSTALL_START_INFO: Started install activateFollowing packages shall be
activated:
/flash/cat9k-wlc.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
/flash/cat9k-webui.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
/flash/cat9k-srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
/flash/cat9k-sipspace.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
/flash/cat9k-sipbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
/flash/cat9k-rpboot.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
/flash/cat9k-rpbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
/flash/cat9k-guestshell.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
/flash/cat9k-espspace.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
/flash/cat9k-cc_srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg

```

```

This operation requires a reload of the system. Do you want to proceed? [y/n]y
--- Starting Activate ---
Performing Activate on all members
[1] Activate package(s) on switch 1
--- Starting list of software package changes ---
Old files list:
  Removed cat9k-cc_srdriver.16.06.02.SPA.pkg
  Removed cat9k-espbase.16.06.02.SPA.pkg
  Removed cat9k-guestshell.16.06.02.SPA.pkg
  Removed cat9k-rpbase.16.06.02.SPA.pkg
  Removed cat9k-rpboot.16.06.02.SPA.pkg
  Removed cat9k-sipbase.16.06.02.SPA.pkg
  Removed cat9k-sipspa.16.06.02.SPA.pkg
  Removed cat9k-srdriver.16.06.02.SPA.pkg
  Removed cat9k-webui.16.06.02.SPA.pkg
  Removed cat9k-wlc.16.06.02.SPA.pkg
New files list:
  Added cat9k-cc_srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
  Added cat9k-espbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
  Added cat9k-guestshell.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
  Added cat9k-rpbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
  Added cat9k-rpboot.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
  Added cat9k-sipbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
  Added cat9k-sipspa.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
  Added cat9k-srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
  Added cat9k-webui.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
  Added cat9k-wlc.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Finished list of software package changes
[1] Finished Activate on switch 1
Checking status of Activate on [1]
Activate: Passed on [1]
Finished Activate

*Oct 30 20:15:56.572: %IOSXE-5-PLATFORM: Switch 1 R0/0: Oct 30 20:15:56 rollback_timer.sh:

%INSTALL-5-INSTALL_AUTO_ABORT_TIMER_PROGRESS: Install auto abort timer will expire in
7200 seconds
Install will reload the system now!
SUCCESS: install_activate Mon Oct 30 20:16:01 UTC 2017

Device#
*Oct 30 20:16:01.935: %IOSXE-5-PLATFORM: Switch 1 R0/0: Oct 30 20:16:01
install_engine.sh: %INSTALL-5-INSTALL_COMPLETED_INFO: Completed install activate PACKAGE

Chassis 1 reloading, reason - Reload command

```

次に示すのは、**show install summary** コマンドがソフトウェアパッケージのステータスをアクティブでありコミット未完了と表示する場合の出力例です。

```

Device# show install summary

[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
             C - Activated & Committed, D - Deactivated & Uncommitted
-----
Type  St  Filename/Version
-----
IMG   I   16.6.2.0
IMG   U   16.6.1.0
Device#

```

次の例では、**install commit** コマンドの実行方法を示しています。

```
Device# install commit
install_commit: START Fri Jun 23 21:24:45 IST 2017
install_commit: Committing PACKAGE

--- Starting Commit ---
Performing Commit on Active/Standby
  [R0] Commit package(s) on R0
  [R0] Finished Commit on R0
Checking status of Commit on [R0]
Commit: Passed on [R0]
Finished Commit

SUCCESS: install_commit  Fri Jun 23 21:24:48 IST 2017

Device#
```

次の例は、更新プログラムパッケージを基本パッケージにロールバックする方法を示しています。

```
Device# install rollback to committed

install_rollback: START Mon Oct 30 20:53:33 UTC 2017

This operation requires a reload of the system. Do you want to proceed? [y/n]

*Oct 30 20:53:34.713: %IOSXE-5-PLATFORM: Switch 1 R0/0: Oct 30 20:53:34
install_engine.sh: %INSTALL-5-INSTALL_START_INFO: Started install rollback

--- Starting Rollback ---
Performing Rollback on all members
  [1] Rollback package(s) on switch 1
    --- Starting rollback impact ---
    Changes that are part of this rollback
    Current   : rp 0 0   rp_boot   cat9k-rpboot.16.06.02.prd9.SPA.pkg
    Current   : rp 1 0   rp_boot   cat9k-rpboot.16.06.02.prd9.SPA.pkg
    Replacement: rp 0 0   rp_boot   cat9k-rpboot.16.06.02.SPA.pkg
    Replacement: rp 1 0   rp_boot   cat9k-rpboot.16.06.02.SPA.pkg
    Current   : cc 0 0   cc_srdriver cat9k-cc_srdriver.16.06.02.prd9.SPA.pkg
    Current   : cc 0 0   cc         cat9k-sipbase.16.06.02.prd9.SPA.pkg
    Current   : cc 0 0   cc_spa     cat9k-sipspa.16.06.02.prd9.SPA.pkg
    Current   : cc 1 0   cc_srdriver cat9k-cc_srdriver.16.06.02.prd9.SPA.pkg
    Current   : cc 1 0   cc         cat9k-sipbase.16.06.02.prd9.SPA.pkg
    Current   : cc 1 0   cc_spa     cat9k-sipspa.16.06.02.prd9.SPA.pkg
    Current   : cc 10 0  cc         cat9k-sipbase.16.06.02.prd9.SPA.pkg
    Current   : cc 10 0  cc_spa     cat9k-sipspa.16.06.02.prd9.SPA.pkg
    Current   : cc 10 0  cc_srdriver cat9k-cc_srdriver.16.06.02.prd9.SPA.pkg
    Current   : cc 2 0   cc_srdriver cat9k-cc_srdriver.16.06.02.prd9.SPA.pkg
    Current   : cc 2 0   cc         cat9k-sipbase.16.06.02.prd9.SPA.pkg
    Current   : cc 2 0   cc_spa     cat9k-sipspa.16.06.02.prd9.SPA.pkg
    Current   : cc 3 0   cc_srdriver cat9k-cc_srdriver.16.06.02.prd9.SPA.pkg
    Current   : cc 3 0   cc         cat9k-sipbase.16.06.02.prd9.SPA.pkg
    Current   : cc 3 0   cc_spa     cat9k-sipspa.16.06.02.prd9.SPA.pkg
    Current   : cc 4 0   cc_srdriver cat9k-cc_srdriver.16.06.02.prd9.SPA.pkg
    Current   : cc 4 0   cc         cat9k-sipbase.16.06.02.prd9.SPA.pkg
    Current   : cc 4 0   cc_spa     cat9k-sipspa.16.06.02.prd9.SPA.pkg
    Current   : cc 5 0   cc_srdriver cat9k-cc_srdriver.16.06.02.prd9.SPA.pkg
    Current   : cc 5 0   cc         cat9k-sipbase.16.06.02.prd9.SPA.pkg
    Current   : cc 5 0   cc_spa     cat9k-sipspa.16.06.02.prd9.SPA.pkg
    Current   : cc 6 0   cc_srdriver cat9k-cc_srdriver.16.06.02.prd9.SPA.pkg
    Current   : cc 6 0   cc         cat9k-sipbase.16.06.02.prd9.SPA.pkg
    Current   : cc 6 0   cc_spa     cat9k-sipspa.16.06.02.prd9.SPA.pkg
```

```

Current      : cc 7 0 cc_srdriver cat9k-cc_srdriver.16.06.02.prd9.SPA.pkg
Current      : cc 7 0 cc cat9k-sipbase.16.06.02.prd9.SPA.pkg
Current      : cc 7 0 cc_spa cat9k-sipspa.16.06.02.prd9.SPA.pkg
Current      : cc 8 0 cc_srdriver cat9k-cc_srdriver.16.06.02.prd9.SPA.pkg
Current      : cc 8 0 cc cat9k-sipbase.16.06.02.prd9.SPA.pkg
Current      : cc 8 0 cc_spa cat9k-sipspa.16.06.02.prd9.SPA.pkg
Current      : cc 9 0 cc_srdriver cat9k-cc_srdriver.16.06.02.prd9.SPA.pkg
Current      : cc 9 0 cc cat9k-sipbase.16.06.02.prd9.SPA.pkg
Current      : cc 9 0 cc_spa cat9k-sipspa.16.06.02.prd9.SPA.pkg
Current      : fp 0 0 fp cat9k-espbase.16.06.02.prd9.SPA.pkg
Current      : fp 1 0 fp cat9k-espbase.16.06.02.prd9.SPA.pkg
Current      : rp 0 0 guestshell cat9k-guestshell.16.06.02.prd9.SPA.pkg
Current      : rp 0 0 rp_base cat9k-rpbase.16.06.02.prd9.SPA.pkg
Current      : rp 0 0 rp_daemons cat9k-rpbase.16.06.02.prd9.SPA.pkg
Current      : rp 0 0 rp_iosd cat9k-rpbase.16.06.02.prd9.SPA.pkg
Current      : rp 0 0 rp_security cat9k-rpbase.16.06.02.prd9.SPA.pkg
Current      : rp 0 0 rp_webui cat9k-webui.16.06.02.prd9.SPA.pkg
Current      : rp 0 0 rp_wlc cat9k-wlc.16.06.02.prd9.SPA.pkg
Current      : rp 0 0 srdriver cat9k-srdriver.16.06.02.prd9.SPA.pkg
Current      : rp 1 0 guestshell cat9k-guestshell.16.06.02.prd9.SPA.pkg
Current      : rp 1 0 rp_base cat9k-rpbase.16.06.02.prd9.SPA.pkg
Current      : rp 1 0 rp_daemons cat9k-rpbase.16.06.02.prd9.SPA.pkg
Current      : rp 1 0 rp_iosd cat9k-rpbase.16.06.02.prd9.SPA.pkg
Current      : rp 1 0 rp_security cat9k-rpbase.16.06.02.prd9.SPA.pkg
Current      : rp 1 0 rp_webui cat9k-webui.16.06.02.prd9.SPA.pkg
Current      : rp 1 0 rp_wlc cat9k-wlc.16.06.02.prd9.SPA.pkg
Current      : rp 1 0 srdriver cat9k-srdriver.16.06.02.prd9.SPA.pkg
Replacement: cc 0 0 cc_srdriver cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 0 0 cc cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 0 0 cc_spa cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 1 0 cc_srdriver cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 1 0 cc cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 1 0 cc_spa cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 10 0 cc cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 10 0 cc_spa cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 10 0 cc_srdriver cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 2 0 cc_srdriver cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 2 0 cc cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 2 0 cc_spa cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 3 0 cc_srdriver cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 3 0 cc cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 3 0 cc_spa cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 4 0 cc_srdriver cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 4 0 cc cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 4 0 cc_spa cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 5 0 cc_srdriver cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 5 0 cc cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 5 0 cc_spa cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 6 0 cc_srdriver cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 6 0 cc cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 6 0 cc_spa cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 7 0 cc_srdriver cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 7 0 cc cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 7 0 cc_spa cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 8 0 cc_srdriver cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 8 0 cc cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 8 0 cc_spa cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 9 0 cc_srdriver cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 9 0 cc cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 9 0 cc_spa cat9k-sipspa.16.06.02.SPA.pkg
Replacement: fp 0 0 fp cat9k-espbase.16.06.02.SPA.pkg
Replacement: fp 1 0 fp cat9k-espbase.16.06.02.SPA.pkg
Replacement: rp 0 0 guestshell cat9k-guestshell.16.06.02.SPA.pkg
Replacement: rp 0 0 rp_base cat9k-rpbase.16.06.02.SPA.pkg

```

```

Replacement:  rp 0 0  rp_daemons  cat9k-rpbase.16.06.02.SPA.pkg
Replacement:  rp 0 0  rp_iosd      cat9k-rpbase.16.06.02.SPA.pkg
Replacement:  rp 0 0  rp_security  cat9k-rpbase.16.06.02.SPA.pkg
Replacement:  rp 0 0  rp_webui    cat9k-webui.16.06.02.SPA.pkg
Replacement:  rp 0 0  rp_wlc      cat9k-wlc.16.06.02.SPA.pkg
Replacement:  rp 0 0  srdriver    cat9k-srdriver.16.06.02.SPA.pkg
Replacement:  rp 1 0  guestshell   cat9k-guestshell.16.06.02.SPA.pkg
Replacement:  rp 1 0  rp_base      cat9k-rpbase.16.06.02.SPA.pkg
Replacement:  rp 1 0  rp_daemons  cat9k-rpbase.16.06.02.SPA.pkg
Replacement:  rp 1 0  rp_iosd      cat9k-rpbase.16.06.02.SPA.pkg
Replacement:  rp 1 0  rp_security  cat9k-rpbase.16.06.02.SPA.pkg

```

Chassis 1 reloading, reason - Reload command

```

Replacement:  rp 1 0  rp_webui    cat9k-webui.16.06.02.SPA.pkg
Replacement:  rp 1 0  rp_wlc      cat9k-wlc.16.06.02.SPA.pkg
Replacement:  rp 1 0  srdriver    cat9k-srdriver.16.06.02.SPA.pkg

```

Finished rollback impact

```

[1] Finished Rollback on switch 1
Checking status of Rollback on [1]
Rollback: Passed on [1]
Finished Rollback

```

```

Install will reload the system now!
SUCCESS: install_rollback  Mon Oct 30 20:54:23 UTC 2017

```

```

Device#
*Oct 30 20:54:23.576: %IOSXE-5-PLATFORM: Switch 1 R0/0: Oct 30 20:54:23
  install_engine.sh: %INSTALL-5-INSTALL_COMPLETED_INFO: Completed install rollback PACKAGE
*Oct 30 20:54:25.416: %STACKMGR-1-RELOAD: Switch 1 R0/0: stack_mgr:
  Reloading due to reason Reload command Oct 30 20:54:31.615 FP0/0: %PMAN-5-EXITACTION:
  Process manager is exiting: reload fp action requested
Oct 30 20:54

```

次に、**install remove inactive** コマンドの出力例を示します。

```
Device# install remove inactive
```

```

install_remove: START Mon Oct 30 19:51:48 UTC 2017
Cleaning up unnecessary package files
  Scanning boot directory for packages ... done.
  Preparing packages list to delete ...
  done.

```

```

The following files will be deleted:
[switch 1]:
/flash/cat9k-cc_srdriver.16.06.02.SPA.pkg
/flash/cat9k-espbase.16.06.02.SPA.pkg
/flash/cat9k-guestshell.16.06.02.SPA.pkg
/flash/cat9k-rpbase.16.06.02.SPA.pkg
/flash/cat9k-rpboot.16.06.02.SPA.pkg
/flash/cat9k-sipbase.16.06.02.SPA.pkg
/flash/cat9k-sipspa.16.06.02.SPA.pkg
/flash/cat9k-srdriver.16.06.02.SPA.pkg
/flash/cat9k-webui.16.06.02.SPA.pkg
/flash/cat9k-wlc.16.06.02.SPA.pkg
/flash/packages.conf

```

```

Do you want to remove the above files? [y/n]y
[switch 1]:
Deleting file flash:cat9k-cc_srdriver.16.06.02.SPA.pkg ... done.
Deleting file flash:cat9k-espbase.16.06.02.SPA.pkg ... done.

```



```

Deleting file flash:cat9k-guestshell.16.06.02.SPA.pkg ... done.
Deleting file flash:cat9k-rpbase.16.06.02.SPA.pkg ... done.
Deleting file flash:cat9k-rpboot.16.06.02.SPA.pkg ... done.
Deleting file flash:cat9k-sipbase.16.06.02.SPA.pkg ... done.
Deleting file flash:cat9k-sipspa.16.06.02.SPA.pkg ... done.
Deleting file flash:cat9k-srdriver.16.06.02.SPA.pkg ... done.
Deleting file flash:cat9k-webui.16.06.02.SPA.pkg ... done.
Deleting file flash:cat9k-wlc.16.06.02.SPA.pkg ... done.
Deleting file flash:packages.conf ... done.
SUCCESS: Files deleted.
--- Starting Post_Remove_Cleanup ---
Performing Post_Remove_Cleanup on all members
  [1] Post_Remove_Cleanup package(s) on switch 1
  [1] Finished Post_Remove_Cleanup on switch 1
Checking status of Post_Remove_Cleanup on [1]
Post_Remove_Cleanup: Passed on [1]
Finished Post_Remove_Cleanup

SUCCESS: install_remove  Mon Oct 30 19:52:25 UTC 2017
Device#

```

次に、**install abort** コマンドの出力例を示します。

```

Device# install abort

/usr/binos/conf/chasutils.sh: line 428: chasfs_is_dominica: readonly function
/usr/binos/conf/chasutils.sh: line 428: chasfs_is_dominica: readonly function
/usr/binos/conf/chasutils.sh: line 428: chasfs_is_dominica: readonly function
/usr/binos/conf/chasutils.sh: line 428: chasfs_is_dominica: readonly function
install_abort: START Mon Oct 30 20:27:32 UTC 2017
install_abort: Abort type PACKAGE subtype NONE smutype NONE

This install abort would require a reload. Do you want to proceed? [y/n]

*Oct 30 20:27:33.189: %INSTALL-5-INSTALL_START_INFO: Switch 1 R0/0: install_engine:
Started install abort
--- Starting Abort ---
Performing Abort on all members
/usr/binos/conf/chasutils.sh: line 428: chasfs_is_dominica: readonly function
/usr/binos/conf/chasutils.sh: line 428: chasfs_is_dominica: readonly function
/usr/binos/conf/chasutils.sh: line 428: chasfs_is_dominica: readonly function
/usr/binos/conf/chasutils.sh: line 428: chasfs_is_dominica: readonly function
/usr/binos/conf/chasutils.sh: line 428: chasfs_is_dominica: readonly function
/usr/binos/conf/chasutils.sh: line 428: chasfs_is_dominica: readonly function
  [1] Abort package(s) on switch 1
    --- Starting rollback impact ---
    Changes that are part of this rollback
    Current      : rp 0 0  rp_boot
cat9k-rpboot.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
    Current      : rp 1 0  rp_boot
cat9k-rpboot.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
    Replacement: rp 0 0  rp_boot          cat9k-rpboot.16.06.02.SPA.pkg
    Replacement: rp 1 0  rp_boot          cat9k-rpboot.16.06.02.SPA.pkg
    Current      : cc 0 0  cc_srdriver
cat9k-cc_srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
    Current      : cc 0 0  cc
cat9k-sipbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
    Current      : cc 0 0  cc_spa
cat9k-sipspa.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
    Current      : cc 1 0  cc_srdriver
cat9k-cc_srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
    Current      : cc 1 0  cc
cat9k-sipbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg

```

## 例：更新プログラムパッケージの管理

```

Current      : cc 1 0 cc_spa
cat9k-sipspa.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 10 0 cc
cat9k-sipbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 10 0 cc_spa
cat9k-sipspa.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 10 0 cc_srdriver
cat9k-cc_srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 2 0 cc_srdriver
cat9k-cc_srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 2 0 cc
cat9k-sipbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 2 0 cc_spa
cat9k-sipspa.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 3 0 cc_srdriver
cat9k-cc_srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 3 0 cc
cat9k-sipbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 3 0 cc_spa
cat9k-sipspa.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 4 0 cc_srdriver
cat9k-cc_srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 4 0 cc
cat9k-sipbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 4 0 cc_spa
cat9k-sipspa.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 5 0 cc_srdriver
cat9k-cc_srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 5 0 cc
cat9k-sipbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 5 0 cc_spa
cat9k-sipspa.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 6 0 cc_srdriver
cat9k-cc_srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 6 0 cc
cat9k-sipbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 6 0 cc_spa
cat9k-sipspa.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 7 0 cc_srdriver
cat9k-cc_srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 7 0 cc
cat9k-sipbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 7 0 cc_spa
cat9k-sipspa.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 8 0 cc_srdriver
cat9k-cc_srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 8 0 cc
cat9k-sipbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 8 0 cc_spa
cat9k-sipspa.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 9 0 cc_srdriver
cat9k-cc_srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 9 0 cc
cat9k-sipbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 9 0 cc_spa
cat9k-sipspa.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : fp 0 0 fp
cat9k-espbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : fp 1 0 fp
cat9k-espbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : rp 0 0 guestshell
cat9k-guestshell.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : rp 0 0 rp_base
cat9k-rpbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg

```

```

Current      : rp 0 0 rp_daemons
cat9k-rpbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : rp 0 0 rp_iosd
cat9k-rpbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : rp 0 0 rp_security
cat9k-rpbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : rp 0 0 rp_webui
cat9k-webui.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : rp 0 0 rp_wlc
cat9k-wlc.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : rp 0 0 srdriver
cat9k-srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : rp 1 0 guestshell
cat9k-guestshell.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : rp 1 0 rp_base
cat9k-rpbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : rp 1 0 rp_daemons
cat9k-rpbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : rp 1 0 rp_iosd
cat9k-rpbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : rp 1 0 rp_security
cat9k-rpbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : rp 1 0 rp_webui
cat9k-webui.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : rp 1 0 rp_wlc
cat9k-wlc.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : rp 1 0 srdriver
cat9k-srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Replacement: cc 0 0 cc_srdriver cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 0 0 cc cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 0 0 cc_spa cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 1 0 cc_srdriver cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 1 0 cc cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 1 0 cc_spa cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 10 0 cc cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 10 0 cc_spa cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 10 0 cc_srdriver cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 2 0 cc_srdriver cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 2 0 cc cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 2 0 cc_spa cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 3 0 cc_srdriver cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 3 0 cc cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 3 0 cc_spa cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 4 0 cc_srdriver cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 4 0 cc cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 4 0 cc_spa cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 5 0 cc_srdriver cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 5 0 cc cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 5 0 cc_spa cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 6 0 cc_srdriver cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 6 0 cc cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 6 0 cc_spa cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 7 0 cc_srdriver cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 7 0 cc cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 7 0 cc_spa cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 8 0 cc_srdriver cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 8 0 cc cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 8 0 cc_spa cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 9 0 cc_srdriver cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 9 0 cc cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 9 0 cc_spa cat9k-sipspa.16.06.02.SPA.pkg
Replacement: fp 0 0 fp cat9k-espbase.16.06.02.SPA.pkg
Replacement: fp 1 0 fp cat9k-espbase.16.06.02.SPA.pkg
Replacement: rp 0 0 guestshell cat9k-guestshell.16.06.02.SPA.pkg

```

```

Replacement:  rp 0 0  rp_base          cat9k-rpbase.16.06.02.SPA.pkg
Replacement:  rp 0 0  rp_daemons       cat9k-rpbase.16.06.02.SPA.pkg
Replacement:  rp 0 0  rp_iosd          cat9k-rpbase.16.06.02.SPA.pkg
Replacement:  rp 0 0  rp_security      cat9k-rpbase.16.06.02.SPA.pkg
Replacement:  rp 0 0  rp_webui        cat9k-webui.16.06.02.SPA.pkg
Replacement:  rp 0 0  rp_wlc          cat9k-wlc.16.06.02.SPA.pkg
Replacement:  rp 0 0  srdriver        cat9k-srdriver.16.06.02.SPA.pkg
Replacement:  rp 1 0  guestshell      cat9k-guestshell.16.06.02.SPA.pkg
Replacement:  rp 1 0  rp_base         cat9k-rpbase.16.06.02.SPA.pkg
Replacement:  rp 1 0  rp_daemons     cat9k-rpbase.16.06.02.SPA.pkg
Replacement:  rp 1 0  rp_iosd        cat9k-rpbase.16.06.02.SPA.pkg
Replacement:  rp 1 0  rp_security     cat9k-rpbase.16.06.02.SPA.pkg
Replacement:  rp 1 0  rp_webui       cat9k-webui.16.06.02.SPA.pkg
Replacement:  rp 1 0  rp_wlc         cat9k-wlc.16.06.02.SPA.pkg
Replacement:  rp 1 0  srdriver       cat9k-srdriver.16.06.02.SPA.pkg
Finished rollback impact
[1] Finished Abort on switch 1
Checking status of Abort on [1]
Abort: Passed on [1]
Finished Abort

```

```

/usr/binos/conf/chasutils.sh: line 428: chasfs_is_dominica: readonly function
[1]: Performing MCU Upgrade Service
/usr/binos/conf/provfunc.sh: line 8792: $!_log_file: ambiguous redirect
SUCCESS: MCU Upgrade Service finished
Install will reload the system now!
SUCCESS: install_abort Mon Oct 30 20:28:21 UTC 2017
/usr/binos/conf/chasutils.sh: line 428: chasfs_is_dominica: readonly function

```

次に、**install activate auto-abort-timer** コマンドの出力例を示します。

```

Device# install activate auto-abort-timer 30

install_activate: START Mon Oct 30 20:42:28 UTC 2017
install_activate: Activating PACKAGE

*Oct 30 20:42:29.149: %IOSXE-5-PLATFORM: Switch 1 R0/0: Oct 30 20:42:29 install_engine.sh:

%INSTALL-5-INSTALL_START_INFO: Started install activateFollowing packages shall be
activated:
/flash/cat9k-wlc.16.06.02.prd9.SPA.pkg
/flash/cat9k-webui.16.06.02.prd9.SPA.pkg
/flash/cat9k-srdriver.16.06.02.prd9.SPA.pkg
/flash/cat9k-sipspace.16.06.02.prd9.SPA.pkg
/flash/cat9k-sipbase.16.06.02.prd9.SPA.pkg
/flash/cat9k-rpboot.16.06.02.prd9.SPA.pkg
/flash/cat9k-rpbase.16.06.02.prd9.SPA.pkg
/flash/cat9k-guestshell.16.06.02.prd9.SPA.pkg
/flash/cat9k-espbase.16.06.02.prd9.SPA.pkg
/flash/cat9k-cc_srdriver.16.06.02.prd9.SPA.pkg

This operation requires a reload of the system. Do you want to proceed? [y/n]y
--- Starting Activate ---
Performing Activate on all members
[1] Activate package(s) on switch 1
--- Starting list of software package changes ---
Old files list:
Removed cat9k-cc_srdriver.16.06.02.SPA.pkg
Removed cat9k-espbase.16.06.02.SPA.pkg
Removed cat9k-guestshell.16.06.02.SPA.pkg
Removed cat9k-rpbase.16.06.02.SPA.pkg
Removed cat9k-rpboot.16.06.02.SPA.pkg
Removed cat9k-sipbase.16.06.02.SPA.pkg

```

```
Removed cat9k-sipspace.16.06.02.SPA.pkg
Removed cat9k-srdriver.16.06.02.SPA.pkg
Removed cat9k-webui.16.06.02.SPA.pkg
Removed cat9k-wlc.16.06.02.SPA.pkg
New files list:
Added cat9k-cc_srdriver.16.06.02.prd9.SPA.pkg
Added cat9k-espbase.16.06.02.prd9.SPA.pkg
Added cat9k-guestshell.16.06.02.prd9.SPA.pkg
Added cat9k-rpbase.16.06.02.prd9.SPA.pkg
Added cat9k-rpboot.16.06.02.prd9.SPA.pkg
Added cat9k-sipbase.16.06.02.prd9.SPA.pkg
Added cat9k-sipspace.16.06.02.prd9.SPA.pkg
Added cat9k-srdriver.16.06.02.prd9.SPA.pkg
Added cat9k-webui.16.06.02.prd9.SPA.pkg
Added cat9k-wlc.16.06.02.prd9.SPA.pkg
Finished list of software package changes
[1] Finished Activate on switch 1
Checking status of Activate on [1]
Activate: Passed on [1]
Finished Activate

*Oct 30 20:43:39.249: %IOSXE-5-PLATFORM: Switch 1 R0/0: Oct 30 20:43:39 rollback_timer.sh:

%INSTALL-5-INSTALL_AUTO_ABORT_TIMER_PROGRESS: Install auto abort timer will expire in
1800 seconds
Install will reload the system now!
SUCCESS: install_activate Mon Oct 30 20:43:44 UTC 2017

Device#
*Oct 30 20:43:44.615: %IOSXE-5-PLATFORM: Switch 1 R0/0: Oct 30 20:43:44 install_engine.sh:

%INSTALL-5-INSTALL_COMPLETED_INFO: Completed install activate PACKAGE
Chassis 1 reloading, reason - Reload command
```

## ソフトウェア インストールの確認

### 手順の概要

1. **enable**
2. **show install log**
3. **show install summary**
4. **show install package *filesystem: filename***
5. **show install active**
6. **show install inactive**
7. **show install committed**
8. **show install uncommitted**

### 手順の詳細

#### ステップ 1 enable

例 :

```
Device> enable
```

特権 EXEC モードを有効にします。

- パスワードを入力します（要求された場合）。

## ステップ2 show install log

例：

```
Device# show install log
```

デバイスの起動以降に実行されたすべてのソフトウェアインストール動作に関する情報を表示します。

```
Device# show install log
```

```
[0|install_op_boot]: START Sun Jun 11 15:01:37 Universal 2017
[0|install_op_boot]: END SUCCESS Sun Jun 11 15:01:44 Universal 2017
[1|install_commit]: START Mon Jun 12 07:27:31 UTC 2017
[1|install_commit(INFO, )]: Releasing transaction lock...
[1|install_commit(CONSOLE, )]: Committing PACKAGE
[remote|install_commit]: START Mon Jun 12 07:28:08 UTC 2017
[remote|install_commit(INFO, )]: Releasing transaction lock...
[remote|install_commit]: END SUCCESS Mon Jun 12 07:28:41 UTC 2017
[1|install_commit(INFO, )]: [1 2 3]: Performing Commit
SUCCESS: Commit finished
[1|install_commit(INFO, )]: install_commit: START Mon Jun 12 07:28:08 UTC 2017
SUCCESS: install_commit Mon Jun 12 07:28:41 UTC 2017
[1|install_commit(INFO, )]: Remote output from switch 2
[1|install_commit(INFO, )]: install_commit: START Mon Jun 12 07:28:12 UTC 2017
SUCCESS: install_commit Mon Jun 12 07:28:44 UTC 2017
[1|install_commit(INFO, )]: install_commit: START Mon Jun 12 07:28:12 UTC 2017
SUCCESS: install_commit Mon Jun 12 07:28:45 UTC 2017
[1|install_commit]: END SUCCESS Mon Jun 12 07:28:47 UTC 2017
```

## ステップ3 show install summary

例：

```
Device# show install summary
```

すべてのメンバ/現場交換可能ユニット（FRU）のイメージのバージョンとそれらに対応するインストール状態に関する情報を表示します。

- このコマンドの出力は、実行した **install** コマンドによって異なります。

```
Device# show install summary
```

```
[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
           C - Activated & Committed, D - Deactivated & Uncommitted
```

```
-----
Type  St  Filename/Version
-----
```

```
IMG   I   16.6.2.0
IMG   C   16.6.1.0
```

```
Device#
```

#### ステップ4 show install package filesystem: filename

例：

```
Device# show install package flash:cat9k_iosxe.16.06.01.SPA.bin
```

指定したソフトウェア インストール パッケージ ファイルに関する情報を表示します。

```
Device# show install package flash:cat9k_iosxe.16.06.01.SPA.bin
```

```
Package: cat9k_iosxe.16.06.01.SPA.bin
Size: 333806196
Timestamp: Sun Jun 11 14:47:23 2017 UTC
Canonical path: /flash/cat9k_iosxe.16.06.01.SPA.bin

Raw disk-file SHA1sum:
 5e9ef6ed1f7472b35eddd61df300e44b14b65ec4
Header size:      1000 bytes
Package type:     10002
Package flags:    0
Header version:   3

Internal package information:
Name: cc_srdriver
BuildTime:
ReleaseDate: Sun-27-Aug-17-09:05
BootArchitecture: none
RouteProcessor: cat9k
Platform: CAT9K
User: mcpre
PackageName: cc_srdriver
Build: BLD_V166_THROTTLE_LATEST_20170827_090555
CardTypes:

Package is not bootable.
Device#
```

#### ステップ5 show install active

例：

```
Device# show install active
```

アクティブなソフトウェア インストール パッケージに関する情報を表示します。

```
Device# show install active
[ R0 ] Active Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
             C - Activated & Committed, D - Deactivated & Uncommitted
-----
Type  St   Filename/Version
-----
IMG   C    16.6.2.0
```

#### ステップ6 show install inactive

例：

```
Device# show install inactive
```

非アクティブなパッケージに関する情報を表示します。

## 例：DHCP サーバーとしてのデバイスの設定

```
Device# show install inactive

[ R0 ] Inactive Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
           C - Activated & Committed, D - Deactivated & Uncommitted
-----
Type  St  Filename/Version
-----
IMG   I   16.7.1.0
Device#
```

## ステップ7 show install committed

例：

```
Device# show install committed
```

コミット済みのパッケージに関する情報を表示します。

```
Device# show install committed
[ R0 ] Committed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
           C - Activated & Committed, D - Deactivated & Uncommitted
-----
Type  St  Filename/Version
-----
IMG   C   16.6.1.0
Device#
```

## ステップ8 show install uncommitted

例：

```
Device# show install uncommitted
```

コミットされていないパッケージに関する情報を表示します。

```
Device# show install uncommitted
[ R0 ] Uncommitted Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
           C - Activated & Committed, D - Deactivated & Uncommitted
-----
Type  St  Filename/Version
-----
IMG   U   16.6.2.0
Device#
```

## 例：DHCP サーバーとしてのデバイスの設定

```
デバイス# configure terminal
デバイス(config)# ip dhcp pool pool1
デバイス(dhcp-config)# network 10.10.10.0 255.255.255.0
デバイス(dhcp-config)# boot config-boot.text
デバイス(dhcp-config)# default-router 10.10.10.1
デバイス(dhcp-config)# option 150 10.10.10.1
```



```
デバイス (dhcp-config) # exit
デバイス (config) # tftp-server flash:config-boot.text
デバイス (config) # interface gigabitethernet1/0/4
デバイス (config-if) # no switchport
デバイス (config-if) # ip address 10.10.10.1 255.255.255.0
デバイス (config-if) # end
```

## 例：DHCP 自動イメージアップデートの設定

```
デバイス # configure terminal
デバイス (config) # ip dhcp pool pool1
デバイス (dhcp-config) # network 10.10.10.0 255.255.255.0
デバイス (dhcp-config) # boot config-boot.text
デバイス (dhcp-config) # default-router 10.10.10.1
デバイス (dhcp-config) # option 150 10.10.10.1
デバイス (dhcp-config) # option 125 hex
0000.0009.0a05.08661.7574.6f69.6e73.7461.6c6c.5f64.686370
デバイス (dhcp-config) # exit
デバイス (config) # tftp-server flash:config-boot.text
デバイス (config) # tftp-server flash:image_name
デバイス (config) # tftp-server flash:boot-config.text
デバイス (config) # tftp-server flash:autoinstall_dhcp
デバイス (config) # interface gigabitethernet1/0/4
デバイス (config-if) # no switchport
デバイス (config-if) # ip address 10.10.10.1 255.255.255.0
デバイス (config-if) # end
```

## 例：DHCP サーバーから設定をダウンロードするためのデバイスの設定

次に、VLAN 99 上のレイヤ 3 SVI インターフェイスを使用し、保存されているコンフィギュレーションで DHCP ベースの自動設定をイネーブルにする例を示します。

```
デバイス # configure terminal
デバイス (config) # boot host dhcp
デバイス (config) # boot host retry timeout 300
デバイス (config) # banner config-save ^C Caution - Saving Configuration File to NVRAM May
Cause You to No longer Automatically Download Configuration Files at Reboot^C
デバイス (config) # vlan 99
デバイス (config-vlan) # interface vlan 99
デバイス (config-if) # no shutdown
デバイス (config-if) # end
デバイス # show boot
BOOT path-list:
Config file:          flash:/config.text
Private Config file: flash:/private-config.text
Enable Break:        no
Manual Boot:         no
```

例：ソフトウェアイメージのリロードのスケジューリング

```
HELPER path-list:
NVRAM/Config file
  buffer size:      32768
Timeout for Config
  Download:         300 seconds
Config Download
  via DHCP:         enabled (next boot: enabled)
デバイス#
```

## 例：ソフトウェアイメージのリロードのスケジューリング

次に、当日の午後 7 時 30 分に、ソフトウェアをdeviceにリロードする例を示します。

```
デバイス# reload at 19:30
Reload scheduled for 19:30:00 UTC Wed Jun 5 2013 (in 2 hours and 25 minutes)
Proceed with reload? [confirm]
```

次に、未来の日時を指定して、ソフトウェアをdeviceにリロードする例を示します。

```
デバイス# reload at 02:00 jun 20
Reload scheduled for 02:00:00 UTC Thu Jun 20 2013 (in 344 hours and 53 minutes)
Proceed with reload? [confirm]
```

## デバイスセットアップの実行に関する追加情報

### 関連資料

関連項目	マニュアルタイトル
デバイスセットアップ コマンド ブート ローダ コマンド	<i>Command Reference (Catalyst 9300 Series Switches)</i>
ハードウェアの設置	<i>Cisco Catalyst 9300 シリーズ スイッチ ハードウェア 設置ガイド</i> .

## デバイスセットアップ設定の実行に関する機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS XE Everest 16.5.1a	デバイスのセットアップ設定	IP アドレス割り当てと DHCP の自動設定を含むデバイスセットアップ設定を実行できます。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、<http://www.cisco.com/go/cfn> [英語] からアクセスします。



## 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。