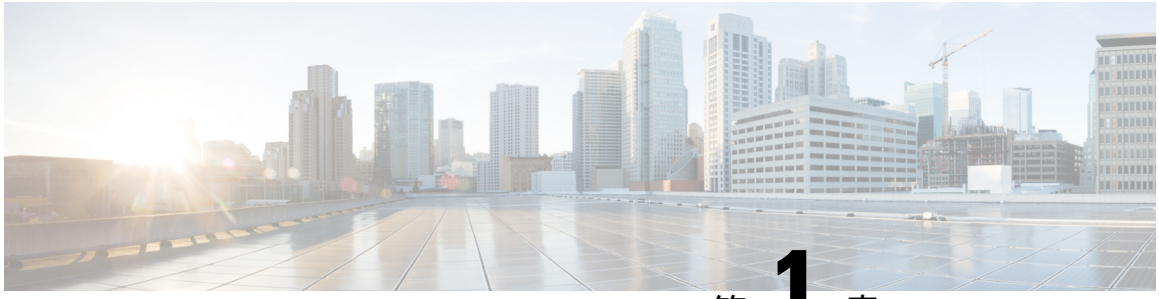
Cisco IOS XE Everest 16.6.x（Catalyst 9500 スイッチ）システム管理コンフィギュレーション ガイド

初版：2017 年 7 月 31 日

最終更新：2017 年 11 月 3 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/

目 次



第 1 章

デバイスの管理

デバイスの管理に関する情報

システム日時の管理

デバイスのシステム日時は自動設定方式（RTC および NTP）または手動設定方式を使用して管理できます。

（注） ここで使用するコマンドの構文および使用方法の詳細については、*Cisco.com* で、『*Cisco IOS Configuration Fundamentals Command Reference*』を参照してください。

システム クロック

時刻サービスの基本となるのはシステム クロックです。このクロックはシステムがスタートアップした瞬間から稼働し、日時を常時トラッキングします。

システム クロックは、次のソースにより設定できます。

- NTP
- 手動設定

システム クロックは、次のサービスに時刻を提供します。

- ユーザの **show** コマンド

- ログおよびデバッグ メッセージ

システム クロックは、グリニッジ標準時（GMT）とも呼ばれる協定世界時（UTC）に基づいて内部的に時刻を追跡します。ローカルのタイムゾーンおよび夏時間に関する情報を設定することにより、時刻がローカルのタイム ゾーンに応じて正確に表示されるようにできます。

システムクロックは、時刻に信頼性があるかどうか（つまり、信頼できると見なされるタイムソースによって時刻が設定されているか）を常時トラッキングします。信頼性のない場合は、時刻は表示目的でのみ使用され、再配信されません。

ネットワーク タイム プロトコル

NTPは、ネットワーク上のデバイス間の時刻の同期化を目的に設計されています。NTPはユーザ データグラム プロトコル（UDP）で稼働し、UDP は IP 上で稼働します。NTP は RFC 1305 で規定されています。

NTP ネットワークは通常、タイム サーバに接続されたラジオ クロックやアトミック クロックなど、正規の時刻源から時刻を取得します。NTPは、この時間をネットワーク全体に配信します。NTP はきわめて効率的で、1 分間に 1 パケットを使用するだけで、2 台のデバイスを 1 ミリ秒以内に同期化できます。

NTP では、信頼できるタイム ソースから各マシンが何 NTP ホップ隔たっているかを表すために、ストラタムという概念が使用されます。ストラタム 1 タイム サーバには、ラジオ クロックまたは原子時計が直接接続されており、ストラタム 2 タイム サーバは、NTP を使用してストラタム 1 タイム サーバから時刻を取得します（以降のストラタムも同様です）。NTP が稼働するデバイスは、タイム ソースとして、NTP を使用して通信するストラタム番号が最小のデバイスを自動的に選択します。この方法によって、NTP時刻配信の自動編成型ツリーが効率的に構築されます。

NTPでは、同期化されていないデバイスと同期化しないことによって、時刻が正確でないデバイスとの同期化を防ぎます。また、NTP では、複数のデバイスから報告される時刻を比較して、ストラタムの番号が小さくても、時刻が他のデバイスと大幅に異なるデバイスとは同期化しません。

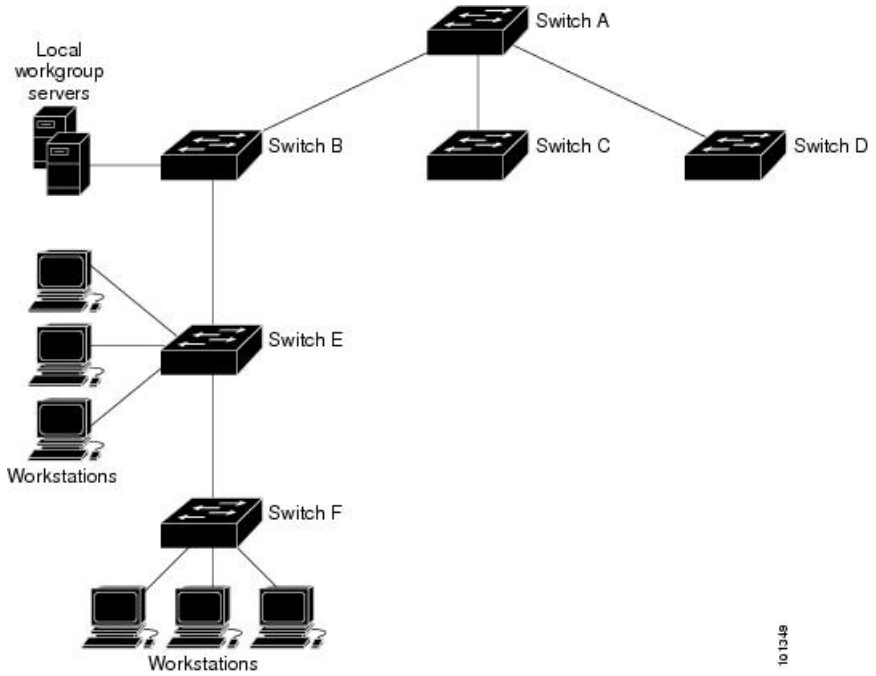
NTPが稼働するデバイス間の通信（アソシエーション）は、通常静的に設定されます。各デバイスには、アソシエーションを作成すべきすべてのデバイスの IP アドレスが与えられます。アソシエーションのペアとなるデバイス間で NTP メッセージを交換することによって、正確な時刻の維持が可能になります。ただし、LAN 環境では、代わりに IP ブロードキャスト メッセージを使用するようにNTPを設定できます。各デバイスを、単にブロードキャストメッセージを送受信するように設定すればよいので、この代替手段によって設定の複雑さが緩和されます。ただし、情報の流れは一方向に限られます。

デバイス上で維持される時刻は、重要なリソースです。NTPのセキュリティ機能を使用して、不正確な時刻が誤って、あるいは意図的に設定されることがないようにしてください。その方法として、アクセス リストベースの制約方式と暗号化認証方式があります。

シスコによる NTP の実装では、ストラタム 1 サービスをサポートしていないため、ラジオ クロックまたは原子時計に接続できません。ネットワークの時刻サービスは、IPインターネット上のパブリック NTP サーバから取得することを推奨します。

次の図に NTP を使用した一般的なネットワークの例を示します。デバイス A は、NTP サーバモードで設定した**デバイス** B、C、D の NTP マスターです。スイッチ B、C、D とデバイス A の間にはサーバ アソシエーションが設定されています。デバイス E はアップストリームおよびダウンストリーム デバイス、デバイス B およびデバイス F それぞれの NTP ピアとして設定されます。

図 *1 :* 一般的な *NTP* ネットワークの構成



ネットワークがインターネットから切り離されている場合、シスコの NTP によって、実際には、他の方法で時刻を学習しているにもかかわらず、デバイスが NTP を使用して同期化しているように動作を設定できます。他のデバイスは、NTP によりこのデバイスと同期化されます。

複数のタイム ソースがある場合は、NTP は常に、より信頼性があると見なされます。NTP の時刻は、他の方法による時刻に優先します。

自社のホスト システムに NTP ソフトウェアを組み込んでいるメーカーが数社あり、UNIX システム用のバージョンやその派生ソフトウェアも一般に入手できます。このソフトウェアによって、ホスト システムも時間が同期化されます。

NTP ストラタム

NTP では、信頼できるタイム ソースから各マシンが何 NTP ホップ隔たっているかを表すために、ストラタムという概念が使用されます。ストラタム 1 タイム サーバには、ラジオ クロックまたは原子時計が直接接続されており、ストラタム 2 タイム サーバは、NTP を使用してストラタム 1 タイム サーバから時刻を取得します（以降のストラタムも同様です）。NTP が稼働するデバイスは、タイム ソースとして、NTP を使用して通信するストラタム番号が最小の

デバイスを自動的に選択します。この方法によって、NTP時刻配信の自動編成型ツリーが効率的に構築されます。

NTPでは、同期化されていないデバイスと同期化しないことによって、時刻が正確でないデバイスとの同期化を防ぎます。また、NTP では、複数のデバイスから報告される時刻を比較して、ストラタムの番号が小さくても、時刻が他のデバイスと大幅に異なるデバイスとは同期化しません。

NTP アソシエーション

NTPが稼働するデバイス間の通信（アソシエーション）は、通常静的に設定されます。各デバイスには、アソシエーションを作成すべきすべてのデバイスの IP アドレスが与えられます。アソシエーションのペアとなるデバイス間で NTP メッセージを交換することによって、正確な時刻の維持が可能になります。ただし、LAN 環境では、代わりに IP ブロードキャスト メッセージを使用するようにNTPを設定できます。各デバイスを、単にブロードキャストメッセージを送受信するように設定すればよいので、この代替手段によって設定の複雑さが緩和されます。ただし、情報の流れは一方向に限られます。

NTP セキュリティ

デバイス上で維持される時刻は、重要なリソースです。NTPのセキュリティ機能を使用して、不正確な時刻が誤って、あるいは意図的に設定されることがないようにしてください。その方法として、アクセス リストベースの制約方式と暗号化認証方式があります。

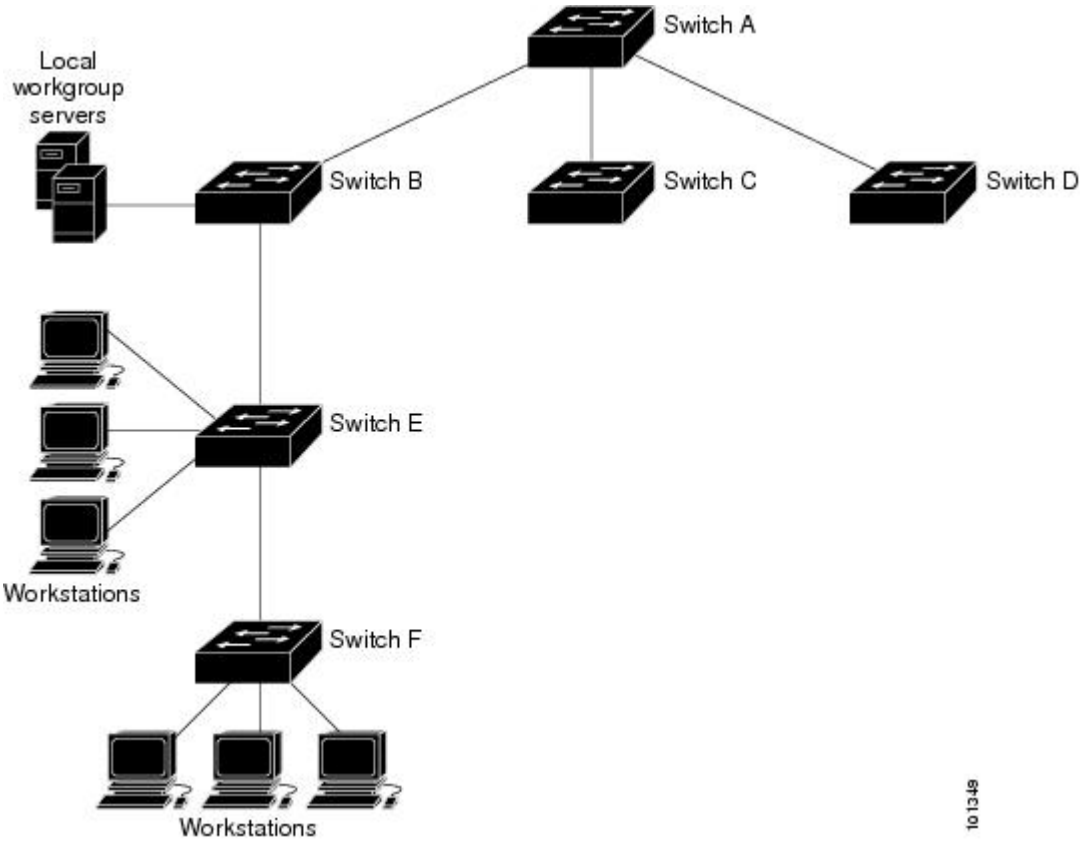
NTP の実装

NTP の実装では、ストラタム 1 サービスがサポートされないため、ラジオ クロックまたは原子時計に接続できません。ネットワークの時刻サービスは、IPインターネット上のパブリック NTP サーバから取得することを推奨します。

図 *2 :* 一般的な *NTP* ネットワークの構成

次の図はNTPを使用した一般的なネットワークの例を示します。スイッチAは、スイッチB、C、D が NTP サーバ モードに設定されている（スイッチ A との間にサーバ アソシエーションが設定されている）場合のNTPマスターです。スイッチEは、アップストリーム スイッチ（ス

イッチ B）とダウンストリーム スイッチ（スイッチ F）の NTP ピアとして設定されます。



ネットワークがインターネットから切り離されている場合、NTPによって、実際には、他の方法で時刻を取得している場合でも、NTPを使用した同期化と同様にデバイスの動作を設定できます。他のデバイスは、NTP によりこのデバイスと同期化されます。

複数のタイム ソースがある場合は、NTP は常に、より信頼性があると見なされます。NTP の時刻は、他の方法による時刻に優先します。

自社のホスト システムに NTP ソフトウェアを組み込んでいるメーカーが数社あり、UNIX システム用のバージョンやその派生ソフトウェアも一般に入手できます。このソフトウェアによって、ホスト システムも時間が同期化されます。

システム名およびシステム プロンプト

デバイスを識別するシステム名を設定します。デフォルトでは、システム名およびプロンプトは *Switch* です。

システム プロンプトを設定していない場合は、システム名の最初の 20 文字がシステム プロンプトとして使用されます。大なり記号（>）が付加されます。システム名が変更されると、プロンプトは更新されます。

ここで使用するコマンドの構文および使用方法の詳細については、『*Cisco IOS Configuration Fundamentals Command Reference*, *Release 12.4*』および『*Cisco IOS IP Command Reference*, *Volume 2 of 3: Routing Protocols*, *Release 12.4*』を参照してください。

デフォルトのシステム名とプロンプトの設定

デフォルトのスイッチのシステム名およびプロンプトは *Switch* です。

DNS

DNS プロトコルは、ドメイン ネーム システム（DNS）を制御します。DNS とは分散型データベースであり、ホスト名を IP アドレスにマッピングできます。デバイスに DNS を設定すると、**ping**、**telnet**、**connect** などのすべての IP コマンドおよび関連する Telnet サポート操作で IP アドレスの代わりにホスト名を使用できます。

IP によって定義される階層型の命名方式では、デバイスを場所またはドメインで特定できます。ドメイン名は、ピリオド（.）を区切り文字として使用して構成されています。たとえば、シスコは、IP で *com* というドメイン名に分類される商業組織なので、ドメイン名は *cisco.com* となります。このドメイン内の特定のデバイス、たとえばファイル転送プロトコル（FTP）システムは、*ftp.cisco.com* で表されます。

IP ではドメイン名をトラッキングするために、ドメイン ネーム サーバという概念が定義されています。ドメイン ネーム サーバの役割は、名前から IP アドレスへのマッピングをキャッシュ（またはデータベース）に保存することです。ドメイン名を IP アドレスにマッピングするには、まず、ホスト名を明示し、ネットワーク上に存在するネーム サーバを指定し、DNS をイネーブルにします。

DNS のデフォルト設定値

表 *1: DNS* のデフォルト設定値

機能	デフォルト設定
DNS イネーブル ステート	有効。
DNS デフォルト ドメイン名	未設定
DNS サーバ	ネーム サーバのアドレスが未設定

ログイン バナー

Message-of-The-Day（MoTD）バナーおよびログイン バナーを作成できます。MOTD バナーはログイン時に、接続されたすべての端末に表示されます。すべてのネットワーク ユーザに影響するメッセージ（差し迫ったシステム シャットダウンの通知など）を送信する場合に便利です。

ログイン バナーも接続されたすべての端末に表示されます。表示されるのは、MoTD バナーの後で、ログイン プロンプトが表示される前です。

（注） ここで使用するコマンドの構文および使用方法の詳細については、『*Cisco IOS Configuration Fundamentals Command Reference, Release 12.4*』を参照してください。

バナーのデフォルト設定

MoTD およびログイン バナーは設定されません。

MAC アドレス テーブル

MAC アドレス テーブルには、デバイスがポート間のトラフィック転送に使用するアドレス情報が含まれています。このアドレス テーブルに登録されたすべての MAC アドレスは、1 つまたは複数のポートに対応しています。アドレス テーブルに含まれるアドレス タイプには、次のものがあります。

- ダイナミック アドレス：デバイスが取得し、使用されなくなった時点で期限切れとなる送信元の MAC アドレス
- スタティック アドレス：手動で入力され、期限切れにならず、デバイスのリセット時にも消去されないユニキャスト アドレス

アドレス テーブルは、宛先 MAC アドレス、対応する VLAN（仮想 LAN）ID、アドレスに対応付けられたポート番号、およびタイプ（スタティックまたはダイナミック）のリストです。

（注） ここで使用するコマンドの構文および使用方法の詳細については、このリリースに対応するコマンド リファレンスを参照してください。

MAC アドレス テーブルの作成

すべてのポートでサポートされる複数の MAC アドレスを使用して、他のネットワーク デバイスにデバイス上のすべてのポートを接続できます。デバイスは、各ポートで受信するパケットの送信元アドレスを取得し、アドレステーブルにアドレスとそれに関連付けられたポート番号を追加することによって、動的なアドレス指定を行います。ネットワークでデバイスの追加または削除が行われると、デバイスによってアドレステーブルが更新され、新しいダイナミックアドレスが追加され、使用されていないアドレスは期限切れになります。

エージング インターバルは、グローバルに設定されています。ただし、デバイスは VLAN ごとにアドレス テーブルを維持し、STP によって VLAN 単位で有効期間を短縮できます。

デバイスは、受信したパケットの宛先アドレスに基づいて、任意の組み合わせのポート間でパケットを送信します。デバイスは、MAC アドレス テーブルを使用することによって、宛先アドレスに関連付けられたポートに限定してパケットを転送します。宛先アドレスがパケットを

送信したポート上にある場合は、パケットはフィルタリング処理され、転送されません。デバイスは、常にストア アンド フォワード方式を使用します。このため、完全なパケットをいったん保存してエラーがないか検査してから転送します。

MAC アドレスおよび VLAN

すべてのアドレスはVLANと関連付けされます。1つのアドレスを複数のVLANに対応付け、それぞれで異なる宛先を設定できます。たとえば、ユニキャスト アドレスを VLAN 1 のポート 1 および VLAN 5 のポート 9、10、1 に転送するといったことが可能です。

VLAN ごとに、独自の論理アドレス テーブルが維持されます。ある VLAN で認識されているアドレスが別の VLAN で認識されるには、別の VLAN 内のポートによって学習されるか、または別の VLAN 内のポートにスタティックに対応付けられる必要があります。

MAC アドレス テーブルのデフォルト設定

次の表に、MAC アドレス テーブルのデフォルト設定を示します。

表 *2: MAC* アドレスのデフォルト設定

機能	デフォルト設定
Aging time	300 秒
ダイナミック アドレス	自動学習
スタティック アドレス	未設定

ARP テーブルの管理

デバイスと通信するには（イーサネット上のデバイスなど）、ソフトウェアは最初にそのデバイスの 48 ビット MAC アドレスまたはローカル データ リンク アドレスを学習する必要があります。IP アドレスからローカル データ リンク アドレスを学習するプロセスを、アドレス解決といいます。

アドレス解決プロトコル（ARP）は、ホスト IP アドレスを、該当するメディアまたは MAC アドレスおよび VLAN ID に対応付けます。IP アドレスを使用して、ARP は対応する MAC アドレスを見つけます。MAC アドレスが見つかると、IP と MAC アドレスとの対応を ARP キャッシュに格納し、すばやく検索できるようにします。その後、IP データグラムがリンク層フレームにカプセル化され、ネットワークを通じて送信されます。イーサネット以外の IEEE 802 ネットワークにおける IP データグラムのカプセル化および ARP 要求/応答については、サブネット ワーク アクセス プロトコル（SNAP）で規定されています。IP インターフェイスでは、標準的なイーサネット形式の ARP カプセル化（**arpa** キーワードで表される）がデフォルトでイネーブルに設定されています。

手動でテーブルに追加された ARP エントリは期限切れにならないので、手動で削除する必要があります。

CLI（コマンドライン インターフェイス）の手順については、*Cisco.com* で Cisco IOS Release 12.4 のマニュアルを参照してください。

デバイスを管理する方法

手動による日付と時刻の設定

正確なシステム時刻は再開と再起動により保持されますが、日付と時刻はシステムが再開してから手動で設定できます。

手動設定は必要な場合にのみ使用することを推奨します。デバイスが同期できる外部ソースがある場合は、システム クロックを手動で設定する必要はありません。

システム クロックの設定

ネットワーク上に、NTP サーバなどの時刻サービスを提供する外部ソースがある場合、手動でシステム クロックを設定する必要はありません。

システム クロックを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	次のいずれかを使用します。 • **clock set** *hh:mm:ss day month year* • **clock set** *hh:mm:ss month day year* 例： `Device# clock set 13:32:00 23 March 2013`	次のいずれかの書式を使ってシステム クロックを手動で設定します。 • *hh:mm:ss*：時間（24 時間形式）、分、秒を指定します。指定された時刻は、設定されたタイム ゾーンに基づきます。 • *day*：月の日で日付を指定します。 • *month*：月を名前で指定します。 • *year*：年を指定します（略式表記で指定しないでください）。

タイム ゾーンの設定

タイム ゾーンを手動で設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	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 3	**clock timezone** *zone hours-offset* [*minutes-offset*] 例： `Device(config)# clock timezone AST -3 30`	時間帯を設定します。 内部時間は、協定世界時（UTC）で維持されるため、このコマンドは表示専用で、時刻を手動で設定するときだけに使用されます。 • *zone*：標準時が適用されているときに表示されるタイム ゾーンの名前を入力します。デフォルトは UTC です。 • *hours-offset*：UTC からのオフセット時間数を入力します。 • （任意）*minutes-offset*：UTC からのオフセット分数を入力します。ローカル タイム ゾーンが UTC と 1 時間の差の割合である場合に指定できます。
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例：	入力を確認します。

	コマンドまたはアクション	目的
	`Device# show running-config`	
ステップ 6	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（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。

夏時間の設定

毎年特定の日に夏時間が開始および終了する地域に夏時間を設定するには、次の作業を行います。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	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 3	**clock summer-time** *zone date month year hh:mm date month year hh:mm* [*offset*]]**date** 例： `Device(config)# clock summer-time PDT date 10 March 2013 2:00 3 November 2013 2:00`	毎年指定された日に開始および終了する夏時間を設定します。
ステップ 4	**clock summer-time** *zone***recurring** [*week day month hh:mm week day month hh:mm* [*offset*]] 例： `Device(config)# clock summer-time`	毎年指定された日に開始および終了する夏時間を設定します。すべての時刻は、現地のタイム ゾーンを基準にしています。開始時間は標準時を基準にしています。

	コマンドまたはアクション	目的
	`PDT recurring 10 March 2013 2:00 3 November 2013 2:00`	終了時間は夏時間を基準にしています。夏時間はデフォルトでディセーブルに設定されています。パラメータなしで **clock summer-time** *zone***recurring** を指定すると、夏時間のルールはデフォルトにより米国のルールになります。 開始月が終了月より後の場合は、システムでは南半球にいると見なされます。 • *zone*：夏時間が有効な場合に表示される時間帯名（PDT など）を指定します。 •（任意）*week*：月の週（1 ～ 4、**first**、または **last**）を指定します。 •（任意）*day*：曜日（Sunday、Monday など）を指定します。 •（任意）*month*：月（January、February など）を指定します。 •（任意）*hh:mm*：時および分単位で時間（24時間形式）を指定します。 •（任意）*offset*：夏時間中に追加する分数を指定します。デフォルトは 60 です。
ステップ 5	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（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。

ユーザの居住地域の夏時間が定期的なパターンに従わない（次の夏時間のイベントの正確な日時を設定する）場合は、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	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 3	**clock summer-time** *zone***date**[*month date year hh:mm month date year hh:mm* [*offset*]]or**clock summer-time** *zone***date** [*date month year hh:mm date month year hh:mm* [*offset*]]	最初の日付で夏時間開始の日付を、2 番めの日付で終了の日付を設定します。 夏時間はデフォルトでディセーブルに設定されています。 • *zone* には、夏時間が施行されているときに表示されるタイム ゾーンの名前（たとえば PDT）を入力します。 • （任意）*week* には、月の何週目かを指定します（1 ～ 5、または last）。 • （任意）*day* には、曜日を指定します（Sunday、Monday など）。 • （任意）*month* には、月を指定します（January、February など）。 • （任意）*hh:mm* には、時刻を時間（24時間形式）と分で指定します。 • （任意）*offset* には、夏時間の間、追加する分数を指定します。デフォルトは 60 です。
ステップ 4	**end** 例：	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
	`Device(config)# end`	
ステップ 5	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（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。

システム名の設定

システム名を手動で設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	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 3	**hostname** 名前 例： `Device(config)# hostname remote-users`	システム名を設定します。システム名を設定すると、システム プロンプトとしても使用されます。 デフォルト設定は Switch です。 名前は ARPANET ホスト名のルールに従う必要があります。このルールではホスト名は文字で始まり、文字または数字で終わり、その間には文字、数字、また

	コマンドまたはアクション	目的
		はハイフンしか使用できません。名前には 63 文字まで使用できます。

ステップ 4	**end**
	例：
	`remote-users(config)#end`
	`remote-users#`
		特権 EXEC モードに戻ります。

ステップ 5	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（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。

DNS の設定

デバイスの IP アドレスをホスト名として使用する場合、この IP アドレスが使用されるため、DNS クエリは発生しません。ピリオド（.）なしでホスト名を設定すると、ピリオドと、それに続くデフォルトのドメイン名がホスト名に追加され、その後で DNS クエリーが行われ、名前を IP アドレスにマッピングします。デフォルトのドメイン名は、グローバル コンフィギュレーション コマンド **ip domain-name** で設定される値です。ホスト名にピリオド（.）がある場合は、Cisco IOS ソフトウェアは、ホスト名にデフォルトのドメイン名を追加せずに IP アドレスを検索します。

DNS を使用するようにスイッチを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 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
	`Device# configure terminal`	
ステップ 3	**ip domain-name** 名前 例： `Device(config)# ip domain-name Cisco.com`	非完全修飾ホスト名（ドット付き 10 進表記ドメイン名のない名前）を完成させるためにソフトウェアが使用する、デフォルトのドメイン名を定義します。 ドメイン名を未修飾の名前から区切るために使用される最初のピリオドは入れないでください。 ブート時にはドメイン名は設定されていませんが、デバイスの設定が BOOTP または Dynamic Host Configuration Protocol（DHCP）サーバから行われている場合、BOOTP または DHCP サーバによってデフォルトのドメイン名が設定されることがあります（この情報がサーバに設定されている場合）。
ステップ 4	**ip name-server** *server-address1* [*server-address2 ... server-address6*] 例： `Device(config)# ip name-server 192.168.1.100 192.168.1.200 192.168.1.300`	名前とアドレスの解決に使用する 1 つまたは複数のネーム サーバのアドレスを指定します。 最大 6 つのネーム サーバを指定できます。各サーバ アドレスはスペースで区切ります。最初に指定されたサーバが、プライマリ サーバです。デバイスは、最初にプライマリ サーバへ DNS クエリを送信します。そのクエリが失敗した場合は、バックアップ サーバにクエリが送信されます。
ステップ 5	**ip domain-lookup** [**nsap** | **source-interface** *interface*] 例： `Device(config)# ip domain-lookup`	（任意） デバイス上で、DNS に基づくホスト名からアドレスへの変換をイネーブルにします。この機能はデフォルトでイネーブルになっています。 ユーザのネットワークデバイスが、名前の割り当てを制御できないネットワーク内のデバイスと接続する必要がある場合、グローバルなインターネットのネーミング方式（DNS）を使用して、ユーザのデバイスを一意に識別するデバイス名を動的に割り当てることができます。

	コマンドまたはアクション	目的
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 8	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（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。

Message-of-the-Day ログイン バナーの設定

デバイスにログインしたときに画面に表示される 1 行以上のメッセージ バナーを作成できます。

MOTD ログイン バナーを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	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 3	**banner motd** *c message c*
	例：
	`Device(config)# banner motd #`
	`This is a secure site. Only`
		MoTD を指定します。
		c：ポンド記号（#）など、目的のデリミタを入力して **Return** キーを押します。区切り文字はバナー テキストの始まり

	コマンドまたはアクション	目的

	`authorized users are allowed.`
	`For access, contact technical support.`
	`#`
		と終わりを表します。終わりの区切り文字の後ろの文字は廃棄されます。
		message：255 文字までのバナー メッセージを入力します。メッセージ内には区切り文字を使用できません。

ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（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。

ログイン バナーの設定

接続されたすべての端末でログイン バナーが表示されるように設定できます。バナーが表示されるのは、MoTD バナーの後で、ログイン プロンプトが表示される前です。

ログイン バナーを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 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
	`Device# configure terminal`	

ステップ 3	**banner login** *c message c*
	例：
	`Device(config)# banner login $`
	`Access for authorized users only.`
	`Please enter your username and password.`
	`$`
		ログイン メッセージを指定します。
		c：ポンド記号（#）など、目的のデリミタを入力して Return キーを押します。区切り文字はバナー テキストの始まりと終わりを表します。終わりの区切り文字の後ろの文字は廃棄されます。
		message：255 文字までのログイン メッセージを入力します。メッセージ内には区切り文字を使用できません。

ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（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。

MAC アドレス テーブルの管理

アドレス エージング タイムの変更

ダイナミック アドレス テーブルのエージング タイムを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	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 3	**mac address-table aging-time** [*0* | *10-1000000*] [**routed-mac** | **vlan** *vlan-id*] 例： `Device(config)# mac address-table aging-time 500 vlan 2`	ダイナミック エントリが使用または更新された後、MAC アドレス テーブル内に保持される時間を設定します。 範囲は 10 ～ 1000000 秒です。デフォルトは 300 です。0 を入力して期限切れをディセーブルにすることもできます。スタティック アドレスは、期限切れになることもテーブルから削除されることもありません。 *vlan-id*：有効な ID は 1 ～ 4094 です。
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（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。

MAC アドレス変更通知トラップの設定

NMSホストにMACアドレス変更通知トラップを送信するようにスイッチを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	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 3	**snmp-server host** *host-addr community-string notification-type* { **informs** | **traps** } {**version** {**1** | **2c** | **3**}} {**vrf** *vrf instance name*} 例： `Device(config)# snmp-server host 172.20.10.10 traps private mac-notification`	トラップ メッセージの受信側を指定します。 • *host-addr*：NMS の名前またはアドレスを指定します。 • **traps** （デフォルト）：ホストに SNMP トラップを送信します。 • **informs**：ホストに SNMP インフォームを送信します。 • **version**：サポートする SNMP バージョンを指定します。informs にはバージョン 1（デフォルト）を使用できません。 • *community-string*：通知処理で送信する文字列を指定します。**snmp-server host** コマンドを使用してこの文字列を設定できますが、この文字列を定義するには、**snmp-server community** コマンドを使用し、次に **snmp-server host** コマンドを使用することを推奨します。 • *notification-type*：**mac-notification** キーワードを使用します。

	コマンドまたはアクション	目的
		• **vrf** *vrf instance name*：このホストの VPN ルーティング/転送インスタンスを指定します。
ステップ 4	**snmp-server enable traps mac-notification change** 例： `Device(config)# snmp-server enable traps mac-notification change`	デバイスが MAC アドレス変更通知トラップを送信できるようにします。
ステップ 5	**mac address-table notification change** 例： `Device(config)# mac address-table notification change`	MACアドレス変更通知機能をイネーブルにします。

ステップ 6	**mac address-table notification change** [**interval** *value*] [**history-size** *value*]
	例：
	`Device(config)# mac address-table notification change interval 123`
	`Device(config)#mac address-table notification change history-size 100`
		トラップ インターバル タイムと履歴テーブルのサイズを入力します。
		• （任意）**interval** *value*：NMS に生成されるトラップの各セット間の通知トラップインターバルを秒単位で指定します。指定できる範囲は 0 ～ 2147483647 秒です。デフォルトは 1 秒です。
		• （任意）**history-size** *value*：MAC 通知履歴テーブルの最大エントリ数を指定します。指定できる範囲は 0 ～ 500 です。デフォルトは 1 です。

ステップ 7	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet1/0/2`	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、SNMP MAC アドレス通知トラップをイネーブルにするレイヤ 2 インターフェイスを指定します。
ステップ 8	**snmp trap mac-notification change** {**added** | **removed**} 例：	インターフェイス上で MAC アドレス変更通知トラップをイネーブルにします。

	コマンドまたはアクション	目的
	`Device(config-if)# snmp trap mac-notification change added`	• MACアドレスがインターフェイスに追加された（**added**）場合にトラップをイネーブルにします。 • MACアドレスがインターフェイスから削除された（**removed**）場合に MAC 通知トラップをイネーブルにします。
ステップ 9	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 10	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 11	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（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。

MAC アドレス移動通知トラップの設定

MAC移動通知を設定する場合は、MACアドレスが、同じVLAN内のあるポートから別のポートに移動すると常に、SNMP 通知が生成されてネットワーク管理システムに送信されます。

NMSホストにMACアドレス移動通知トラップを送信するようにデバイスを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。

	コマンドまたはアクション	目的
ステップ 2	**configureterminal** 例： `Device# configure terminal`	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 3	**snmp-server host** *host-addr* {**traps** | **informs**} {**version** {**1** | **2c** | **3**}} *community-string notification-type* 例： `Device(config)# snmp-server host 172.20.10.10 traps private mac-notification`	トラップ メッセージの受信側を指定します。 • *host-addr*：NMS の名前またはアドレスを指定します。 • **traps** （デフォルト）：ホストに SNMP トラップを送信します。 • **informs**：ホストに SNMP インフォームを送信します。 • **version**：サポートする SNMP バージョンを指定します。informs にはバージョン 1（デフォルト）を使用できません。 • *community-string*：通知処理で送信する文字列を指定します。**snmp-server host** コマンドを使用してこの文字列を設定できますが、この文字列を定義するには、**snmp-server community** コマンドを使用し、次に **snmp-server host** コマンドを使用することを推奨します。 • *notification-type*：**mac-notification** キーワードを使用します。
ステップ 4	**snmp-server enable traps mac-notification move** 例： `Device(config)# snmp-server enable traps mac-notification move`	デバイスが NMS に MAC アドレス移動通知トラップを送信できるようにします。
ステップ 5	**mac address-table notification mac-move** 例： `Device(config)# mac address-table`	MAC アドレス移動通知機能をイネーブルにします。

	コマンドまたはアクション	目的
	`notification mac-move`	
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 8	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（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。

次のタスク

スイッチによる MAC アドレス移動通知トラップの送信をディセーブルにするには、**no snmp-server enable traps mac-notification move** グローバル コンフィギュレーション コマンドを使用します。MAC アドレス変更通知機能をディセーブルにするには、**no mac address-table notification mac-move** グローバル コンフィギュレーション コマンドを使用します。

設定を確認するには、**show mac address-table notification mac-move** 特権 EXEC コマンドを入力します。

MAC しきい値通知トラップの設定

MAC しきい値通知を設定する場合は、MAC アドレス テーブルのしきい値の制限値に達するか、その値を超えると、SNMP 通知が生成されてネットワーク管理システムに送信されます。

NMS ホストに MAC アドレス テーブルしきい値通知トラップを送信するようにスイッチを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。

	コマンドまたはアクション	目的
ステップ 2	**configureterminal** 例： `Device# configure terminal`	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 3	**snmp-server host** *host-addr* {**traps** | **informs**} {**version** {**1** | **2c** | **3**}} *community-string notification-type* 例： `Device(config)# snmp-server host 172.20.10.10 traps private mac-notification`	トラップ メッセージの受信側を指定します。 • *host-addr*：NMS の名前またはアドレスを指定します。 • **traps** （デフォルト）：ホストに SNMP トラップを送信します。 • **informs**：ホストに SNMP インフォームを送信します。 • **version**：サポートする SNMP バージョンを指定します。informs にはバージョン 1（デフォルト）を使用できません。 • *community-string*：通知処理で送信する文字列を指定します。**snmp-server host** コマンドを使用してこの文字列を設定できますが、この文字列を定義するには、**snmp-server community** コマンドを使用し、次に **snmp-server host** コマンドを使用することを推奨します。 • *notification-type*：**mac-notification** キーワードを使用します。
ステップ 4	**snmp-server enable traps mac-notification threshold** 例： `Device(config)# snmp-server enable traps mac-notification threshold`	NMS への MAC しきい値通知トラップをイネーブルにします。
ステップ 5	**mac address-table notification threshold** 例： `Device(config)# mac address-table`	MAC アドレスしきい値通知機能をイネーブルにします。

	コマンドまたはアクション	目的
	`notification threshold`	

ステップ 6	**mac address-table notification threshold** [**limit** *percentage*] | [**interval** *time*]
	例：
	`Device(config)# mac address-table notification threshold interval 123`
	`Device(config)# mac address-table notification threshold limit 78`
		MAC アドレスしきい値使用状況モニタリングのしきい値を入力します。
		• （任意）**limit** *percentage*：MAC アドレス テーブルの使用率を指定します。有効値は 1 ～ 100 % です。デフォルト値は 50% です。
		• （任意）**interval** *time*：通知の間隔を指定します。有効値は120秒以上です。デフォルトは 120 秒です。

ステップ 7	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 8	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 9	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（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。

スタティック アドレス エントリの追加および削除

スタティック アドレスを追加するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。

	コマンドまたはアクション	目的
ステップ 2	**configureterminal** 例： `Device# configure terminal`	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 3	**mac address-table static** *mac-addr***vlan** *vlan-id***interface** *interface-id* 例： `Device(config)# mac address-table static c2f3.220a.12f4 vlan 4 interface gigabitethernet 1/0/1`	MAC アドレス テーブルにスタティック アドレスを追加します。 • *mac-addr*：アドレス テーブルに追加する宛先 MAC ユニキャスト アドレスを指定します。この宛先アドレスを持つパケットが指定した VLAN に着信すると、指定したインターフェイスに転送されます。 • *vlan-id*：指定された MAC アドレスを持つパケットを受信する VLAN を指定します。指定できる VLAN ID の範囲は 1 ～ 4094 です。 • *interface-id*：受信パケットが転送されるインターフェイスを指定します。有効なインターフェイスは、物理ポートまたはポート チャネルです。スタティック マルチキャスト アドレスの場合、複数のインターフェイス ID を入力できます。スタティック ユニキャスト アドレスの場合、インターフェイスは同時に 1 つしか入力できません。ただし、同じ MAC アドレスおよび VLAN ID を指定すると、コマンドを複数回入力できます。
ステップ 4	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 5	**copy running-config startup-config** 例： `Device# copy running-config`	（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。

	コマンドまたはアクション	目的
	startup-config	

ユニキャスト MAC アドレス フィルタリングの設定

デバイスが送信元または宛先ユニキャスト スタティック アドレスをドロップするよう設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	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 3	**mac address-table static** *mac-addr* **vlan** *vlan-id* **drop** 例： `Device(config)# mac address-table static c2f3.220a.12f4 vlan 4 drop`	ユニキャスト MAC アドレス フィルタリングをイネーブルにし、デバイスが指定した送信元または宛先ユニキャスト スタティック アドレスを持つパケットをドロップするように設定します。 • *mac-addr*：送信元または宛先ユニキャスト MAC アドレス（48 ビット）を指定します。この MAC アドレスを持つパケットはドロップされます。 • *vlan-id*：指定された MAC アドレスを持つパケットを受信する VLAN を指定します。指定できる VLAN ID の範囲は 1 ～ 4094 です。
ステップ 4	**end** 例： `Device(config)# end`	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 5	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（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。

デバイスのモニタリングおよび保守の管理

コマンド	目的
clear mac address-table dynamic	すべてのダイナミックエントリを削除します。
clear mac address-table dynamic address *mac-address*	特定の MAC アドレスを削除します。
clear mac address-table dynamic interface *interface-id*	指定された物理ポートまたはポート チャネル上のすべてのアドレスを削除します。
clear mac address-table dynamic vlan *vlan-id*	指定された VLAN 上のすべてのアドレスを削除します。
show clock [*detail*]	時刻と日付の設定を表示します。
show ip igmp snooping groups	すべての VLAN または指定された VLAN に対するレイヤ 2 マルチキャスト エントリを表示します。
show mac address-table address *mac-address*	指定された MAC アドレスの MAC アドレス テーブル情報を表示します。
show mac address-table aging-time	すべての VLAN または指定された VLAN のエージング タイムを表示します。
show mac address-table count	すべての VLAN または指定された VLAN で存在しているアドレス数を表示します。
show mac address-table dynamic	ダイナミック MAC アドレス テーブル エントリのみを表示します。

コマンド	目的
show mac address-table interface *interface-name*	指定されたインターフェイスのMACアドレステーブル情報を表示します。
show mac address-table move update	MACアドレステーブル移動更新情報を表示します。
show mac address-table multicast	マルチキャストのMACアドレスのリストを表示します。
show mac address-table notification {change | mac-move | threshold}	MAC通知パラメータおよび履歴テーブルを表示します。
show mac address-table secure	セキュア MAC アドレスを表示します。
show mac address-table static	スタティック MAC アドレス テーブル エントリだけを表示します。
show mac address-table vlan *vlan-id*	指定された VLAN の MAC アドレス テーブル情報を表示します。

デバイス管理の設定例

例：システム クロックの設定

次の例は、システム クロックを手動で設定する方法を示しています。

```
Device# clock set 13:32:00 23 July 2013
```

例：サマー タイムの設定

次に、サマータイムが 3 月 10 日の 02:00 に開始し、11 月 3 日の 02:00 に終了する場合の設定を例として示します。

```
Device(config)# clock summer-time PDT recurring PST date
10 March 2013 2:00 3 November 2013 2:00
```

次に、サマータイムの開始日と終了日を設定する例を示します。

```
Device(config)#clock summer-time PST date
20 March 2013 2:00 20 November 2013 2:00
```

例：MOTD バナーの設定

次の例は、開始および終了デリミタにポンド記号（#）を使用して、MOTD バナーを設定する方法を示しています。

```
Device(config)# banner motd #

This is a secure site. Only authorized users are allowed.
For access, contact technical support.

#

Device(config)#
```

次に、前の設定により表示されたバナーの例を示します。

```
Unix> telnet 192.0.2.15

Trying 192.0.2.15...

Connected to 192.0.2.15.

Escape character is '^]'.

This is a secure site. Only authorized users are allowed.

For access, contact technical support.

User Access Verification

Password:
```

例：ログイン バナーの設定

次の例は、開始および終了デリミタにドル記号（$）を使用して、にログイン バナーを設定する方法を示しています。

```
Device(config)# banner login $

Access for authorized users only. Please enter your username and password.

$

Device(config)#
```

例：MAC アドレス変更通知トラップの設定

次に、NMS として 172.20.10.10 を指定し、NMS への MAC アドレス通知トラップの送信をイネーブルにし、MAC アドレス変更通知機能をイネーブルにし、インターバル タイムを 123 秒

に設定し、履歴サイズを 100 エントリに設定し、特定のポートで MAC アドレスが追加された場合のトラップをイネーブルにする例を示します。

```
Device(config)# snmp-server host 172.20.10.10 traps private mac-notification
Device(config)# snmp-server enable traps mac-notification change
Device(config)# mac address-table notification change
Device(config)# mac address-table notification change interval 123
Device(config)# mac address-table notification change history-size 100
Device(config)# interface gigabitethernet1/2/1
Device(config-if)# snmp trap mac-notification change added
```

例：MAC しきい値通知トラップの設定

次に、NMS として 172.20.10.10 を指定し、MAC アドレスしきい値通知機能をイネーブルにし、インターバル タイムを 123 秒に設定し、制限を 78% に設定する例を示します。

```
Device(config)# snmp-server host 172.20.10.10 traps private mac-notification
Device(config)# snmp-server enable traps mac-notification threshold
Device(config)# mac address-table notification threshold
Device(config)# mac address-table notification threshold interval 123
Device(config)# mac address-table notification threshold limit 78
```

例：MAC アドレス テーブルへのスタティック アドレスの追加

次の例では、MAC アドレス テーブルにスタティック アドレス c2f3.220a.12f4 を追加する方法を示します。VLAN 4 でこの MAC アドレスを宛先アドレスとして持つパケットを受信すると、パケットは指定されたポートに転送されます。

（注） 複数のインターフェイスに同じ静的 MAC アドレスを関連付けることはできません。コマンドを別のインターフェイスで再度実行すると、新しいインターフェイス上で静的 MAC アドレスが上書きされます。

```
Device(config)# mac address-table static c2f3.220a.12f4 vlan 4 interface
gigabitethernet1/1/1
```

例：ユニキャスト MAC アドレス フィルタリングの設定

次に、ユニキャスト MAC アドレス フィルタリングをイネーブルにし、c2f3.220a.12f4 の送信元または宛先アドレスを持つドロップパケットを設定する例を示します。送信元または宛先としてこの MAC アドレスを持つパケットが VLAN4 上で受信された場合、パケットがドロップされます。

```
Device(config)# mac address-table static c2f3.220a.12f4 vlan 4 drop
```

デバイス管理に関する追加情報

関連資料

関連項目	参照先
システム管理コマンド	*Command Reference (Catalyst 9500 Series Switches)*

MIB

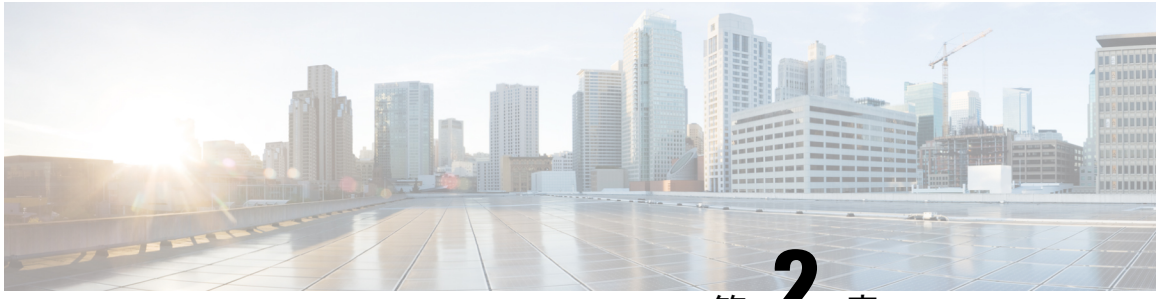
MIB	MIB リンク
本リリースでサポートするすべての MIB	選択したプラットフォーム、Cisco IOS リリース、およびフィーチャ セットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

テクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Product Alert Tool（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

デバイス管理の機能履歴と情報

リリース	変更箇所
Cisco IOS XE Everest 16.5.1a	この機能が導入されました。



第 2 章

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

- 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。
- アクティブとスタンバイのルータ プロセッサ（RT）間での自動アップグレード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デバイス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

デバイスを起動するには、ハードウェア設置ガイドの手順に従ってデバイスを設置して電源投入し、デバイス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、デバイスを起動します。

ブート ローダにより、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 ローダの使用目的は通常、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。

デバイス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 フォーマットをデバイスのコンソール 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

デバイスでは、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 モードでデバイスを起動できます:

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

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、デバイス 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。

インストール モードでの起動とは異なり、バンドル モードでの起動は複数のメディアから利用できます：

- flash:
- usbflash0:
- tftp:

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

ソフトウェア パッケージをデバイスに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 モードで実行します。

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 ロケーション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。

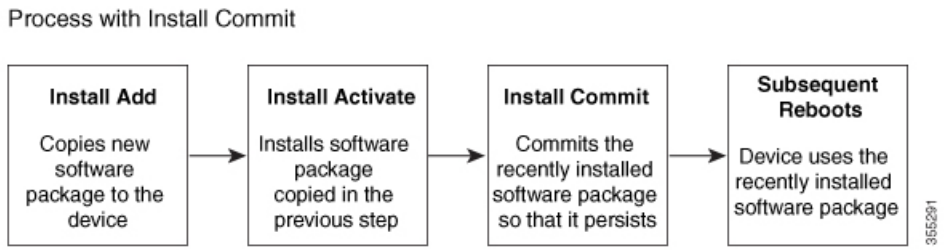
install activate コマンドを動作させるには、パッケージをデバイスのブートフラッシュで使用可能にする必要があります。このコマンドを設定すると、.bin ファイルから以前に追加したパッケージがアクティブ化され、システムがリロードします。

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 *3:* ソフトウェア パッケージのコミット



ソフトウェア インストールの中止

ソフトウェア イメージのアクティブ化は次の方法で中止できます。

- **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中止されます。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。

  install auto-abort-timer stop コマンドを使用して、このタイマーを停止します。

- **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。**install commit** コマンドを発行する前に、このコマンドを使用します。

デバイス 情報の割り当て

IP 情報を割り当てるには、デバイスのセットアップ プログラムを使用する方法、Dynamic Host Configuration Protocol（DHCP）サーバを使用する方法、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、デバイスのセットアップ 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。

サーバの設定後は DHCP サーバを使用して、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、デバイス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応答しないでください。

デバイスの設定手順を熟知している経験豊富なユーザの場合は、デバイス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 *3:* 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。

機能	デフォルト設定
デフォルト ゲートウェイ（Default gateway）	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。
ホストネーム	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。デバイスは、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、デバイス（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、デバイス上で 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。

デバイスの DHCP サーバは、スイッチと同じ LAN 上に配置することも、そのデバイスとは別の 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、デバイスと 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す。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。

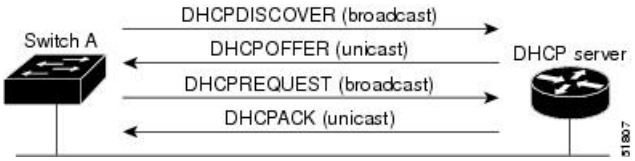
DHCP ベースの自動設定は、デバイスの BOOTP クライアント機能に代わるものです。

DHCP クライアントの要求プロセス

デバイスを起動したときに、デバイス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 *4 : 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。デバイス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す。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。デバイスが BOOTP サーバからの応答を受け入れ、自身を設定する場合、デバイスはデバイス 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、デバイス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 サーバから取得できます。クライアント（デバイス）は DCHPDISCOVER メッセージ内に、DHCP サーバ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 つ以上のデバイスに新しいイメージ 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デバイスが、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 サーバからネットワーク内の 1 つ以上のデバイス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、デバイス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、デバイス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 つ以上のデバイス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つのデバイススイッチ（または複数のデバイス）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）。

デバイスの 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 ホスト名）、オプション 150（TFTP サーバ 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。

デバイス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デバイス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デバイスにインストールされます。デバイスを再起動すると、このコンフィギュレーションがデバイス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サーバには、デバイスのハードウェア アドレスによって各デバイス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。
- デバイスに IP アドレス情報を受信させるには、DHCP サーバに次のリース 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（デバイスで使用するデフォルト ゲートウェイ アドレス）（必須）
- デバイスにTFTPサーバ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 サーバの設定によっては、デバイスは 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、デバイスは設定されません。ルータの IP アドレスまたは TFTP サーバ名が見つからなかった場合、デバイスは 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。
- デバイスは DHCP サーバとして動作可能です。デフォルトでは、Cisco IOS DHCP サーバおよび DHCP リレー エージェント機能はデバイス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 サーバの設定に基づいて、デバイスは TFTP サーバから 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 サーバへの IP 接続に必要なすべてのオプションについてデバイスに応答するよう 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 サーバを設定している場合、デバイスは指定された TFTP サーバ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、デバイスはファイル名と TFTP サーバ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* はデバイスの現在のホスト名です。使用される TFTP サーバアドレスには、（存在する場合）指定された TFTP サーバのアドレス、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。

デバイス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（実際のデバイス コンフィギュレーション ファイル）。
- network-confg 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-confg または ciscortr.cfg ファイル（これらのファイルには、すべてのデバイス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 サーバが、デバイスとは異なる LAN 上にある場合、またはデバイス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、デバイス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、デバイスと同じ 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、デバイスはルータを介して 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、デバイス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、デバイス用に予約され、DHCP 応答（1 ファイル読み込み方式）で提供されている場合

  デバイスは 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。デバイスは、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。

- デバイスの 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）。

  デバイスは 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。デバイスは、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 アドレスだけがデバイス用に予約され、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）

  デバイス 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。デバイスは、TFTP サーバにユニキャストメッセージを送信し、network-confg 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-confg ファイルが読み込めない場合、デバイスは 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、デバイスのホスト名から IP アドレスへのマッピングが含まれています。デバイスは、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、デバイスは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、デバイスはデフォルトのスイッチを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、デバイス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-confg または cisconet.cfg のどちらが先に読み込まれたかに応じて、*hostname*-confg または *hostname*.cfg）を TFTP サーバから読み込みます。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-confg、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、デバイスは router-confg ファイルを読み込みます。router-confg ファイルを読み込むことができない場合、デバイスは ciscortr.cfg ファイルを読み込みます。

（注） DHCP 応答から TFTP サーバ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 サーバ名を IP アドレスに変換できない場合には、デバイスは TFTP サーバ要求をブロードキャストします。

環境変数の制御方法

通常動作デバイスでは、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 ローダ モードを開始します。電源コードを再接続中にデバイス電源コードを取り外し、[Mode] ボタンを押します。システム LED が緑色に点滅してから点灯状態になったら、[Mode] ボタンを放します。ブートローダの デバイス プロンプトが表示されます。

デバイスのブート 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。通常の環境では、環境変数の設定を変更する必要はありません。

一般的な環境変数

この表では、最も一般的な環境変数の機能について説明します。

表 *4*：一般的な環境変数

変数	ブートローダ コマンド	Cisco IOS グローバル コンフィギュレーション コマンド
BOOT	**set BOOT** *filesystem*:/ *file-url* ... 自動起動時にロードして実行を試みる、セミコロンで区切られた実行可能ファイルのリスト。	**boot system** {*filesystem*: /*file-url* ...| **switch** {*number* | **all**}} 次回の起動時にロードする Cisco IOS イメージ、および、を指定します。このコマンドは、BOOT 環境変数の設定を変更します。 パッケージプロビジョニング ファイルは、*packages.conf* ファイルとも呼ばれ、起動時にどのソフトウェアパッケージをアクティブ化するかを判断するために、システムが使用するものです。 • インストールモードで起動する場合、アクティブ化するパッケージを指定するために、**boot** コマンドで指定されたパッケージプロビジョニング ファイルが使用されます。例：**boot flash:packages.conf**。 • バンドルモードで起動する場合、起動したバンドルに含まれているパッケージのプロビジョニング ファイルがバンドルに含まれているパッケージのアクティブ化に使用されます。例：**boot flash:image.bin**。

変数	ブートローダ コマンド	Cisco IOS グローバル コンフィギュレーション コマンド
MANUAL_BOOT	**set MANUAL_BOOT yes** スイッチの起動を自動で行うか手動で行うかを決定します。 有効な値は 1、yes、0、および no です。no または 0 に設定されている場合、ブートローダはシステムを自動的に起動しようとします。それ以外の値に設定されている場合は、ブート ローダ モードから手動でスイッチを起動する必要があります。	**boot manual** 次回の起動時にスイッチを手動で起動できるようにします。MANUAL_BOOT 環境変数の設定が変更されます。 次回のシステム再起動時には、スイッチはブートローダ モードになります。システムを起動するには、**boot flash:** *filesystem***:/** *file-url* ブート ローダ コマンドを使用してブート可能なイメージの名前を指定します。
CONFIG_FILE	**set CONFIG_FILE flash:/** *file-url* Cisco IOS がシステム コンフィギュレーションの不揮発性コピーの読み書きに使用するファイル名を変更します。	**boot config-file flash:/** *file-url* Cisco IOS がシステム設定の不揮発性コピーの読み書きに使用するファイル名を指定します。このコマンドによって、CONFIG_FILE 環境変数が変更されます。
BAUD	**set BAUD** *baud-rate*	**line console 0** **speed** *speed-value* ボー レートを設定します。
ENABLE_BREAK	**set ENABLE_BREAK yes/no**	**boot enable-break switch yes/no** 自動起動時の break をイネーブルにします。**break** コマンドの入力に与えられた時間は 5 秒です。

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 *5 : TFTP* の環境変数

変数	説明
MAC_ADDR	スイッチの MAC アドレスを指定します。 （注） 変数は変更しないことを推奨します。 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。
IP_ADDRESS	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。
DEFAULT_ROUTER	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。

ソフトウェア イメージのリロードのスケジューリング

デバイス上でソフトウェアイメージのリロードを後で（深夜や週末など、デバイスをあまり使用しないときに）行うよう、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デバイスでソフトウェアのアップグレードを実行する場合などに）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デバイス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、デバイスがブート ローダ モードになることでリモート ユーザが制御を失う、ということ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デバイス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 を使用してデバイス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 つのデバイス を設定する必要があります。1 つ目のデバイスは DHCP サーバおよび TFTP サーバと同じように機能し、2 つ目のデバイス（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、新しいデバイス。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デバイスで TFTP や DHCP 設定の DHCP 自動設定を行う方法を示します。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	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	**ip dhcp pool** *poolname* 例： `Device(config)# ip dhcp pool pool`	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。

	コマンドまたはアクション	目的
ステップ 3	**boot** *filename* 例： `Device(dhcp-config)# boot config-boot.text`	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	**network** *network-number mask prefix-length* 例： `Device(dhcp-config)# network 10.10.10.0 255.255.255.0`	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	**default-router** *address* 例： `Device(dhcp-config)# default-router 10.10.10.1`	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	**option 150** *address* 例： `Device(dhcp-config)# option 150 10.10.10.1`	TFTP サーバの IP アドレスを指定します。
ステップ 7	**exit** 例： `Device(dhcp-config)# exit`	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 8	**tftp-server flash**:*filename.text* 例： `Device(config)# tftp-server flash:config-boot.text`	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。

	コマンドまたはアクション	目的
ステップ 9	**interface** *interface-id* 例： `Device(config)# interface fortygigabitethernet1/0/4`	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	**no switchport** 例： `Device(config-if)# no switchport`	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	**ip address** *address mask* 例： `Device(config-if)# ip address 10.10.10.1 255.255.255.0`	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	**end** 例： `Device(config-if)# end`	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デバイスで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

最初にデバイス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。テキスト ファイルに、ダウンロードするイメージの名前を指定します（たとえば、c3750e-ipservices-mz.122-44.3.SE.tarc3750x-ipservices-mz.122-53.3.SE2.tar）。このイメージは、bin ファイルでなく、tar ファイルである必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 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
	`Device# configure terminal`	
ステップ 2	**ip dhcp pool** *poolname* 例： `Device(config)# ip dhcp pool pool1`	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	**boot** *filename* 例： `Device(dhcp-config)# boot config-boot.text`	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。
ステップ 4	**network** *network-number mask prefix-length* 例： `Device(dhcp-config)# network 10.10.10.0 255.255.255.0`	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	**default-router** *address* 例： `Device(dhcp-config)# default-router 10.10.10.1`	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	**option 150** *address* 例： `Device(dhcp-config)# option 150 10.10.10.1`	TFTP サーバの IP アドレスを指定します。
ステップ 7	**option 125** *hex* 例：	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。

	コマンドまたはアクション	目的
	`Device(dhcp-config)# option 125 hex 0000.0009.0a05.08661.7574.6f69.6e73.7461.6c6c.5f64.686370`	
ステップ 8	**copy tftp flash** *filename.txt* 例： `Device(config)# copy tftp flash image.bin`	デバイスに、テキストファイルをアップロードします。
ステップ 9	**copy tftp flash** *imagename.bin* 例： `Device(config)# copy tftp flash image.bin`	デバイスに、新しいイメージの tar ファイルをアップロードします。
ステップ 10	**exit** 例： `Device(dhcp-config)# exit`	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 11	**tftp-server flash:** *config.text* 例： `Device(config)# tftp-server flash:config-boot.text`	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	**tftp-server flash:** *imagename.bin* 例： `Device(config)# tftp-server flash:image.bin`	TFTP サーバ上のイメージ名を指定します。
ステップ 13	**tftp-server flash:** *filename.txt* 例： `Device(config)# tftp-server flash:boot-config.text`	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。

	コマンドまたはアクション	目的
ステップ 14	**interface** *interface-id* 例： `Device(config)# interface gigabitEthernet1/0/4`	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	**no switchport** 例： `Device(config-if)# no switchport`	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	**ip address** *address mask* 例： `Device(config-if)# ip address 10.10.10.1 255.255.255.0`	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	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。
ステップ 18	**copyrunning-configstartup-config** 例： `Device(config-if)# end`	（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

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 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
	`Device# configure terminal`	
ステップ 2	**boot host dhcp** 例： `Device(conf)# boot host dhcp`	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	**boot host retry timeout** *timeout-value* 例： `Device(conf)# boot host retry timeout 300`	（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。
ステップ 4	**banner config-save ^C** *warning-message* **^C** 例： `Device(conf)# banner config-save ^C Caution - Saving Configuration File to NVRAM May Cause You to No longer Automatically Download Configuration Files at Reboot^C`	（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	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show boot** 例： `Device# show boot`	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

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	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 vlan** *vlan-id* 例： `Device(config)# interface vlan 99`	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	**ip address** *ip-address subnet-mask* 例： `Device(config-vlan)# ip address 10.10.10.2 255.255.255.0`	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	**exit** 例： `Device(config-vlan)# exit`	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	**ip default-gateway** *ip-address* 例： `Device(config)# ip default-gateway 10.10.10.1`	デバイス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、デバイススイッチから宛先 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、デバイスは、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デバイスを設定した場合、デフォルト ゲートウェイの設定は不要です。

	コマンドまたはアクション	目的
		（注） デフォルトゲートウェイの構成に基づいて、デバイスの CAPWAP は中継を行い、ルーティングされたアクセス ポイントとデバイスの接続をサポートします。
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show interfaces vlan** *vlan-id* 例： `Device# show interfaces vlan 99`	設定された IP アドレスを確認します。
ステップ 8	**show ip redirects** 例： `Device# show ip redirects`	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 デバイス を使用します。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 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
	`Switch# configure terminal`	
ステップ 2	**boot flash**:*/file-url* 例： `Switch(config)# boot flash:config.text`	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。 *file-url*：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	**end** 例： `Switch(config)# end`	特権 EXEC モードに戻ります。
ステップ 4	**show boot** 例： `Switch# show boot`	入力を確認します。 **boot** 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	**copyrunning-configstartup-config** 例： `Switch# copy running-config startup-config`	（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

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	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	**boot manual** 例： `Device(config)# boot manual`	次回の起動時に、スイッチを手動で起動できるようにします。
ステップ 3	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 4	**show boot** 例： `Device# show boot`	入力を確認します。 **boot manual** グローバル コンフィギュレーション コマンドによって、MANUAL_BOOT 環境変数の設定が変更されます。 次回、システムを再起動したときには、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 *switch:* 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、**boot** *filesystem:/file-url* ブート ローダ コマンドを使用します。 • *filesystem*：システム ボードのフラッシュ デバイスに flash: を使用します。 `Switch: boot flash:` • *file-url*：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。 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 5	**copyrunning-configstartup-config** 例： `Device# copy running-config startup-config`	（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。

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。次に、**install add file activate commit** 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するタスクを示します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**install add file tftp:** *filename* [**activate commit**] 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。
ステップ 3	**exit** 例： `Device# exit`	特権 EXEC モードを終了し、ユーザ 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

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。

	コマンドまたはアクション	目的
ステップ 2	**install add file tftp:** *filename* 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。 • 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。
ステップ 3	**install activate** [**auto-abort-timer**] 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。 • 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。 • **auto-abort-timer** 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。 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。 **install commit** 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中止されます。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	**install abort** 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中止し、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。 • 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。
ステップ 5	**install commit** 例：	リロードが繰り返されても持続する変更を行います。

	コマンドまたはアクション	目的
	`Device# install commit`	• **install commit** 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。自動アボート 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。
ステップ 6	**install rollback to committed** 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	**install remove** {**file** *filesystem: filename* | **inactive**} 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	**show install summary** 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。 • このコマンドの出力は、設定されている **install** コマンドに応じて変化します。

Deviceをバンドル モードで起動する場合

デバイスを起動するには、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デバイス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、デバイス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 デバイス を起動する方法を示します。

手順

	コマンドまたはアクション	目的
ステップ 1	**switch:BOOT=<source path of .bin file>** 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	**boot** 例： `switch: boot`	デバイスをブートします。
ステップ 3	**show version**	デバイスが**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デバイスを設定する方法について説明します。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	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	**copy running-config startup-config** 例： `copy running-config startup-config`	**reload** コマンドを使用する前に、デバイス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	**reload in** [*hh*:]*mm* [*text*]
	例：
	`Device(config)# reload in 12`
	`System configuration has been modified. Save? [yes/no]: y`
		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。

ステップ 4	**reload at** *hh*: *mm* [*month day* | *day month*] [*text*] 例： `Device(config)# reload at 14:00`	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） デバイスのシステム クロックが（ネットワーク タイムプロトコル（NTP）、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にのみ、**at** キーワードを使用します。時刻は、デバイスに設定されたタイム ゾーンに基づきます。リロードが複数のデバイスで同時に行われるようにスケジューリングするには、各デバイスの時間が NTP と同期している必要があります。
ステップ 5	**reload cancel** 例：	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。

	コマンドまたはアクション	目的
	`Device(config)# reload cancel`	
ステップ 6	**show reload** 例： **show reload**	以前デバイス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.

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
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** boot 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。

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 tftp://172.16.0.1//tftpboot/folder1/
cat9k_iosxe.16.06.01.SPA.bin
install_add: START Fri Jun 23 21:07:59 IST 2017
install_add: Adding PACKAGE

--- Starting Add ---
Performing Add on Active/Standby
issu_helper.sh (13338): drop_caches: 3
  [R0] Add package(s) on R0
  [R0] Finished Add on R0
Checking status of Add on [R0]
Add: Passed on [R0]
Finished Add

SUCCESS: install_add  Fri Jun 23 21:09:10 IST 2017
Device#
```

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 **showinstallsummary** 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 Fri Jun 23 21:13:25 IST 2017
install_activate: Activating PACKAGE
ISOFS: Unable to identify CD-ROM format.
Following packages shall be activated:
/flash/cat9k-webui.BLD_V166_THROTTLE_LATEST_20170622_152342.SSA.pkg
/flash/cat9k-srdriver.BLD_V166_THROTTLE_LATEST_20170622_152342.SSA.pkg
/flash/cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170622_152342.SSA.pkg
/flash/cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170622_152342.SSA.pkg
/flash/cat9k-rpboot.BLD_V166_THROTTLE_LATEST_20170622_152342.SSA.pkg
/flash/cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170622_152342.SSA.pkg
/flash/cat9k-guestshell.BLD_V166_THROTTLE_LATEST_20170622_152342.SSA.pkg
/flash/cat9k-espbase.BLD_V166_THROTTLE_LATEST_20170622_152342.SSA.pkg
/flash/cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170622_152342.SSA.pkg

This operation requires a reload of the system. Do you want to proceed? [y/n]y
--- Starting Activate ---
Performing Activate on Active/Standby
  [R0] Activate package(s) on R0
    --- Starting list of software package changes ---
    Old files list:
      Removed cat9k-cc_srdriver.BLD_POLARIS_DEV_LATEST_20170622_233647.SSA.pkg
      Removed cat9k-espbase.BLD_POLARIS_DEV_LATEST_20170622_233647.SSA.pkg
      Removed cat9k-guestshell.BLD_POLARIS_DEV_LATEST_20170622_233647.SSA.pkg
      Removed cat9k-rpbase.BLD_POLARIS_DEV_LATEST_20170622_233647.SSA.pkg
      Removed cat9k-rpboot.BLD_POLARIS_DEV_LATEST_20170622_233647.SSA.pkg
      Removed cat9k-sipbase.BLD_POLARIS_DEV_LATEST_20170622_233647.SSA.pkg
      Removed cat9k-sipspa.BLD_POLARIS_DEV_LATEST_20170622_233647.SSA.pkg
      Removed cat9k-srdriver.BLD_POLARIS_DEV_LATEST_20170622_233647.SSA.pkg
      Removed cat9k-webui.BLD_POLARIS_DEV_LATEST_20170622_233647.SSA.pkg
    New files list:
      Added cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170622_152342.SSA.pkg
      Added cat9k-espbase.BLD_V166_THROTTLE_LATEST_20170622_152342.SSA.pkg
      Added cat9k-guestshell.BLD_V166_THROTTLE_LATEST_20170622_152342.SSA.pkg
      Added cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170622_152342.SSA.pkg
      Added cat9k-rpboot.BLD_V166_THROTTLE_LATEST_20170622_152342.SSA.pkg
      Added cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170622_152342.SSA.pkg
      Added cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170622_152342.SSA.pkg
      Added cat9k-srdriver.BLD_V166_THROTTLE_LATEST_20170622_152342.SSA.pkg
      Added cat9k-webui.BLD_V166_THROTTLE_LATEST_20170622_152342.SSA.pkg
    Finished list of software package changes
  [R0] Finished Activate on R0
Checking status of Activate on [R0]
Activate: Passed on [R0]
Finished Activate

Install will reload the system now!

Device#
```

次の **showinstallsummary** コマンドの出力例では、ソフトウェア パッケージのステータスがアクティブでありコミット未完了と表示されています。

```
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   U    16.6.1.0
Device#
```

次の例では、**installcommit** コマンドの実行方法を示します。

```
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 Tue Jun 20 14:55:12 PDT 2017

 This operation requires a reload of the system. Do you want to proceed? [y/n]
*Jun 20 14:55:12.911 PDT: %IOSXE-5-PLATFORM:  R0/0: Jun 20 14:55:12 install_engine.sh:
 %INSTALL-5-INSTALL_START_INFO: Started install rollbacky
--- Starting Rollback ---
Performing Rollback on Active/Standby
  [R0] Rollback package(s) on R0
    --- Starting rollback impact ---
    Changes that are part of this rollback
    Current    :  rp 0 0   rp_boot
cat9k-rpboot.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :  rp 1 0   rp_boot
cat9k-rpboot.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Replacement:  rp 0 0   rp_boot
cat9k-rpboot.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:  rp 1 0   rp_boot
cat9k-rpboot.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Current    :   cc 0  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 0 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 0 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
```

```
    Current    :   cc 1  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 1 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 1 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 10 0  cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 10 0  cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 10 0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 2  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 2 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 2 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 3  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 3 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 3 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 4  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 4 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 4 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 5  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 5 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 5 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 6  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 6 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 6 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 7  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 7 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 7 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 8  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 8 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 8 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 9  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 9 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 9 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   fp 0 0   fp
cat9k-espbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   fp 1 0   fp
cat9k-espbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
```

```
    Current    :   rp 0 0   guestshell
cat9k-guestshell.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   rp 0 0   rp_base
cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   rp 0 0   rp_daemons
cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   rp 0 0   rp_iosd
cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   rp 0 0   rp_security
cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   rp 0 0   rp_webui
cat9k-webui.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   rp 0 0   srdriver
cat9k-srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   rp 1 0   guestshell
cat9k-guestshell.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   rp 1 0   rp_base
cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   rp 1 0   rp_daemons
cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   rp 1 0   rp_iosd
cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   rp 1 0   rp_security
cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   rp 1 0   rp_webui
cat9k-webui.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   rp 1 0   srdriver
cat9k-srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Replacement:   cc 0  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 0 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 0 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 1  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 1 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 1 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 10 0  cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 10 0  cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 10 0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 2  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 2 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 2 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 3  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 3 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 3 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 4  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 4 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 4 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
```

```
    Replacement:   cc 5  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 5 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 5 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 6  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 6 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 6 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 7  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 7 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 7 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 8  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 8 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 8 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 9  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 9 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 9 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   fp 0 0   fp
cat9k-espbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   fp 1 0   fp
cat9k-espbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   rp 0 0   guestshell
cat9k-guestshell.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   rp 0 0   rp_base
cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   rp 0 0   rp_daemons
cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   rp 0 0   rp_iosd
cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   rp 0 0   rp_security
cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   rp 0 0   rp_webui
cat9k-webui.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   rp 0 0   srdriver
cat9k-srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   rp 1 0   guestshell
cat9k-guestshell.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   rp 1 0   rp_base
cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   rp 1 0   rp_daemons
cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   rp 1 0   rp_iosd
cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   rp 1 0   rp_security
cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   rp 1 0   rp_webui
cat9k-webui.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   rp 1 0   srdriver
cat9k-srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Finished rollback impact
  [R0] Finished Rollback on R0
```

```
Checking status of Rollback on [R0]
Rollback: Passed on [R0]
Finished Rollback


Install will reload the system now!
SUCCESS: install_rollback  Tue Jun 20 14:56:54 PDT 2017

Device#
```

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 Tue Jun 20 14:14:40 PDT 2017
Cleaning up unnecessary package files
No path specified, will use booted path flash:packages.conf
Cleaning flash:
  Scanning boot directory for packages ... done.
  Preparing packages list to delete ...
    cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
      File is in use, will not delete.
    cat9k-espbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
      File is in use, will not delete.
    cat9k-guestshell.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
      File is in use, will not delete.
    cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
      File is in use, will not delete.
    cat9k-rpboot.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
      File is in use, will not delete.
    cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
      File is in use, will not delete.
    cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
      File is in use, will not delete.
    cat9k-srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
      File is in use, will not delete.
    cat9k-webui.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
      File is in use, will not delete.
    packages.conf
      File is in use, will not delete.
  done.


The following files will be deleted:
[R0]:
/flash/cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
/flash/cat9k-espbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
/flash/cat9k-guestshell.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
/flash/cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
/flash/cat9k-rpboot.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
/flash/cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
/flash/cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
/flash/cat9k-srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
/flash/cat9k-webui.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
/flash/cat9k_1.bin
/flash/cat9k_1.conf
/flash/cat9k_2.1.conf
/flash/cat9k_2.bin
/flash/cat9k_2.conf
/flash/cat9k_iosxe.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.bin
/flash/packages.conf.00-
```

```
Do you want to remove the above files? [y/n]y
[R0]:
Deleting file flash:cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
 ... done.
Deleting file flash:cat9k-espbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg ...
 done.
Deleting file flash:cat9k-guestshell.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
 ... done.
Deleting file flash:cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg ...
 done.
Deleting file flash:cat9k-rpboot.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg ...
 done.
Deleting file flash:cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg ...
 done.
Deleting file flash:cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg ...
 done.
Deleting file flash:cat9k-srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
... done.
Deleting file flash:cat9k-webui.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg ...
done.
Deleting file flash:cat9k_1.bin ... done.
Deleting file flash:cat9k_1.conf ... done.
Deleting file flash:cat9k_2.1.conf ... done.
Deleting file flash:cat9k_2.bin ... done.
Deleting file flash:cat9k_2.conf ... done.
Deleting file flash:cat9k_iosxe.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.bin ...
done.
Deleting file flash:packages.conf.00- ... done.
SUCCESS: Files deleted.
--- Starting Post_Remove_Cleanup ---
Performing Post_Remove_Cleanup on Active/Standby
  [R0] Post_Remove_Cleanup package(s) on R0
  [R0] Finished Post_Remove_Cleanup on R0
Checking status of Post_Remove_Cleanup on [R0]
Post_Remove_Cleanup: Passed on [R0]
Finished Post_Remove_Cleanup


SUCCESS: install_remove  Tue Jun 20 14:16:29 PDT 2017
Device#
```

次に、**install abort** コマンドの出力例を示します。

```
Device# install abort

install_abort: START Tue Jun 20 14:06:48 PDT 2017
install_abort: Abort type PACKAGE

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

*Jun 20 14:06:49.820 PDT:%IOSXE-5-PLATFORM:  R0/0: Jun 20 14:06:49 install_engine.sh:
 %INSTALL-5-INSTALL_START_INFO: Started install aborty
--- Starting Abort ---
Performing Abort on Active/Standby
  [R0] Abort package(s) on R0
    --- Starting rollback impact ---
    Changes that are part of this rollback
    Current    :  rp 0 0   rp_boot
cat9k-rpboot.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :  rp 1 0   rp_boot
cat9k-rpboot.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Replacement:  rp 0 0   rp_boot
```

```
cat9k-rpboot.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:  rp 1 0   rp_boot
cat9k-rpboot.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Current    :   cc 0  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 0 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 0 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 1  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 1 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 1 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 10 0  cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 10 0  cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 10 0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 2  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 2 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 2 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 3  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 3 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 3 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 4  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 4 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 4 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 5  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 5 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 5 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 6  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 6 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 6 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 7  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 7 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 7 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 8  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 8 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 8 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 9  0  cc_srdriver
```

```
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 9 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   cc 9 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   fp 0 0   fp
cat9k-espbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   fp 1 0   fp
cat9k-espbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   rp 0 0   guestshell
cat9k-guestshell.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   rp 0 0   rp_base
cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   rp 0 0   rp_daemons
cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   rp 0 0   rp_iosd
cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   rp 0 0   rp_security
cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   rp 0 0   rp_webui
cat9k-webui.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   rp 0 0   srdriver
cat9k-srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   rp 1 0   guestshell
cat9k-guestshell.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   rp 1 0   rp_base
cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   rp 1 0   rp_daemons
cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   rp 1 0   rp_iosd
cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   rp 1 0   rp_security
cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   rp 1 0   rp_webui
cat9k-webui.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Current    :   rp 1 0   srdriver
cat9k-srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Replacement:   cc 0  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 0 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 0 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 1  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 1 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 1 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 10 0  cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 10 0  cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 10 0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 2  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 2 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 2 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 3  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 3 0   cc
```

```
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 3 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 4  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 4 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 4 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 5  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 5 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 5 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 6  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 6 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 6 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 7  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 7 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 7 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 8  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 8 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 8 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 9  0  cc_srdriver
cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 9 0   cc
cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   cc 9 0   cc_spa
cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   fp 0 0   fp
cat9k-espbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   fp 1 0   fp
cat9k-espbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   rp 0 0   guestshell
cat9k-guestshell.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   rp 0 0   rp_base
cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   rp 0 0   rp_daemons
cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   rp 0 0   rp_iosd
cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   rp 0 0   rp_security
cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   rp 0 0   rp_webui
cat9k-webui.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   rp 0 0   srdriver
cat9k-srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   rp 1 0   guestshell
cat9k-guestshell.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   rp 1 0   rp_base
cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   rp 1 0   rp_daemons
cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   rp 1 0   rp_iosd
```

```
cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   rp 1 0   rp_security
cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   rp 1 0   rp_webui
cat9k-webui.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Replacement:   rp 1 0   srdriver
cat9k-srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    Finished rollback impact
  [R0] Finished Abort on R0
Checking status of Abort on [R0]
Abort: Passed on [R0]
Finished Abort


Install will reload the system now!
SUCCESS: install_abort  Tue Jun 20 14:08:25 PDT 2017

Device#
```

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 Tue Jun 20 17:26:07 PDT 2017
install_activate: Activating PACKAGE


*Jun 20 17:26:08.572 PDT: %IOSXE-5-PLATFORM:  R0/0: Jun 20 17:26:08 install_engine.sh:
 %INSTALL-5-INSTALL_START_INFO: Started install activate


Following packages shall be activated:
/flash/cat9k-webui.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
/flash/cat9k-srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
/flash/cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
/flash/cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
/flash/cat9k-rpboot.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
/flash/cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
/flash/cat9k-guestshell.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
/flash/cat9k-espbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
/flash/cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg


This operation requires a reload of the system. Do you want to proceed? [y/n]
*Jun 20 18:07:47.821 PDT: %ENVIRONMENTAL-6-NOTICE: Temp: DopplerD, Location: R0, State:
 Minor, Reading: 85 Celsius
*Jun 20 18:13:47.848 PDT: %ENVIRONMENTAL-6-NOTICE: Temp:    inlet, Location: R0, State:
 Minor, Reading: 46 Celsiusy
--- Starting Activate ---
Performing Activate on Active/Standby
  [R0] Activate package(s) on R0
    --- Starting list of software package changes ---
    Old files list:
      Removed cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
      Removed cat9k-espbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
      Removed cat9k-guestshell.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
      Removed cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
      Removed cat9k-rpboot.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
      Removed cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
      Removed cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
      Removed cat9k-srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
      Removed cat9k-webui.BLD_V166_THROTTLE_LATEST_20170618_152248.SSA.pkg
    New files list:
      Added cat9k-cc_srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
```

```
      Added cat9k-espbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
      Added cat9k-guestshell.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
      Added cat9k-rpbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
      Added cat9k-rpboot.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
      Added cat9k-sipbase.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
      Added cat9k-sipspa.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
      Added cat9k-srdriver.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
      Added cat9k-webui.BLD_V166_THROTTLE_LATEST_20170618_152248_2.SSA.pkg
    Finished list of software package changes
  [R0] Finished Activate on R0
Checking status of Activate on [R0]
Activate: Passed on [R0]
Finished Activate



*Jun 20 18:53:02.320 PDT: %IOSXE-5-PLATFORM:  R0/0: Jun 20 18:53:02 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  Tue Jun 20 18:53:27 PDT 2017
Device#
```

ソフトウェア インストールの確認

手順

ステップ 1 enble

例：

```
Device> enable
```

特権 EXEC モードをイネーブルにします。

- プロンプトが表示されたら、パスワードを入力します。

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
Device#
```

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。

```
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 サーバとしてのデバイスの設定

```
Device# configure terminal
Device(config)# ip dhcp pool pool1
Device(dhcp-config)# network 10.10.10.0 255.255.255.0
Device(dhcp-config)# boot config-boot.text
Device(dhcp-config)# default-router 10.10.10.1
Device(dhcp-config)# option 150 10.10.10.1
Device(dhcp-config)# exit
Device(config)# tftp-server flash:config-boot.text
Device(config)# interface gigabitethernet1/0/4
Device(config-if)# no switchport
Device(config-if)# ip address 10.10.10.1 255.255.255.0
Device(config-if)# end
```

例：DHCP 自動イメージ アップデートの設定

```
Device# configure terminal
Device(config)# ip dhcp pool pool1
Device(dhcp-config)# network 10.10.10.0 255.255.255.0
```

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

例：DHCP サーバ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。

```
Device# configure terminal
Device(config)# boot host dhcp
Device(config)# boot host retry timeout 300
Device(config)# banner config-save ^C Caution - Saving Configuration File to NVRAM May
Cause You to No longer Automatically Download Configuration Files at Reboot^C
Device(config)# vlan 99
Device(config-vlan)# interface vlan 99
Device(config-if)# no shutdown
Device(config-if)# end
Device# show boot
BOOT path-list:
Config file:          flash:/config.text
Private Config file:  flash:/private-config.text
Enable Break:         no
Manual Boot:          no
HELPER path-list:
NVRAM/Config file
      buffer size:    32768
Timeout for Config
          Download:    300 seconds
Config Download
       via DHCP:       enabled (next boot: enabled)
Device#
```

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、ソフトウェアをデバイスにリロードする例を示します。

```
Device# 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、ソフトウェアをデバイスにリロードする例を示します。

```
Device# 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]
```

デバイスの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	参照先
デバイス セットアップ コマンド ブート ローダ コマンド	*Command Reference (Catalyst 9500 Series Switches)*
ハードウェアの設置	*Cisco Catalyst 9500* シリーズ スイッチ ハードウェア設置ガイド。

標準および RFC

標準/RFC	タイトル
なし	—

MIB

MIB	MIB リンク
本リリースでサポートするすべての MIB	選択したプラットフォーム、Cisco IOS リリース、およびフィーチャ セットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

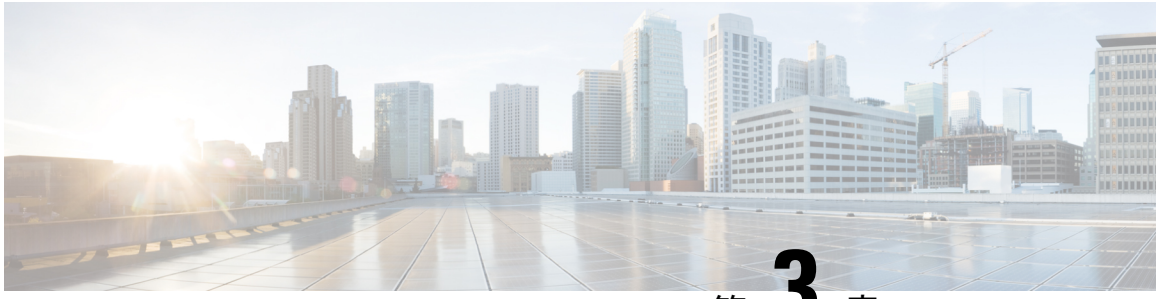
テクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Product Alert Tool（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

デバイスセットアップ設定の機能履歴と情報

コマンド履歴

リリース	変更箇所
Cisco IOS XE Everest 16.5.1a	この機能が導入されました。



第 3 章

Right-To-Use ライセンスの設定

RTU ライセンスの設定に関する制約事項

次に、RTU ライセンスの設定および使用に関する制約事項を示します。

- ライセンスをアクティブ化するには、新しいライセンス レベルを設定した後にスイッチを再起動する必要があります。
- 期限切れの評価ライセンスは、再起動後は再アクティブ化できません。

RTU ライセンスの設定に関する情報

Right-To-Use ライセンス

Right-To-Use（RTU）ライセンスで使用可能なソフトウェア機能は、基本またはアドオンのライセンス レベルに分類されます。使用可能なライセンス タイプは次のとおりです。

- 基本ライセンス：永久ライセンスとして次を注文できます。
  - Network Essentials
  - Network Advantage（Network Essentials も含む）

- アドオン ライセンス：3 年、5 年、および 7 年の固定期間にわたって次のライセンスをサブスクライブできます。
  - Digital Networking Architecture（DNA）Essentials
  - DNA Advantage（DNA Essentials も含む）

  アドオン ライセンスの当初期間が終了した後は、アドオン ライセンスを非アクティブにし、デバイスをリロードすることによって、基本ライセンスを引き続き使用できます。

ライセンスをアクティブにするには、エンドユーザライセンス契約（EULA）に同意し、デバイスをリブートする必要があります。

Cisco Smart Software Manager（CSSM）は、エンタープライズ全体でのライセンスに関わる手続きを簡略化し、シスコ ソフトウェアの購入、導入、追跡、および更新を簡単にします。単一のユーザインターフェイスを通じて、ライセンスの所有権や使用状況を目に見える形にします。

プラットフォーム サポートに関する情報を検出し、機能を使用できるライセンス レベルを確認するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、http://www.cisco.com/go/cfnに進みます。Cisco.com のアカウントは必要ありません。

RTU ライセンスは、次の期間に従って使用できます。

- 永久ライセンス：特定の機能を備え、有効期限のないライセンスを購入できます。あるデバイスから別のデバイスへ移行できます。
- 期間ライセンス：特定のサブスクリプション期間に機能セットとともに購入できます。Cisco Smart Software Manager（CSSM）から有効期限を確認できます。
- 評価ライセンス：デバイスに事前にインストールされており、90 日間試用できます。このライセンスは購入できません。また、別のデバイスに移行することもできません。ライセンスが有効になると、このタイプのライセンスは期限が切れるまで非アクティブ化できません。評価ライセンスの期限切れに関する警告システム メッセージが、90 日の有効期限が切れる 10 日前と 5 日前に生成され、90日目以降は毎日生成されます。評価期間が終了すると、次のリロードの際にデバイスはそのデフォルトのライセンスに戻りますが、ネットワークの運用には影響しません。

基本ライセンスとともにアドオンライセンスを購入する場合、許可されている組み合わせと、許可されていない組み合わせに注意してください。

表 *6*：ライセンスの組み合わせ

組み合わせ	ライセンス レベル（**License Level**）
許可されているライセンスの組み合わせ	Network Essentials + DNA Essentials
	Network Advantage + DNA Essentials
	Network Advantage + DNA Advantage

（注） DNA Advantage ライセンスは Network Essentials の基本ライセンスに追加できません。

RTU ライセンスの設定方法

ライセンスの有効化

手順

	コマンドまたはアクション	目的
ステップ 1	**license right-to-use** [**activate** | **deactivate**] [**network-essentials** | **network-advantage**] [**all** | **evaluation** {**all** | **slot** *slot-number <1-8>*}] [**acceptEULA**] 例： `Device# license right-to-use activate network-essentials all acceptEULA`	ライセンス レベルをアクティブにします。すべてのスイッチ上でアクティブ化され、EULA への同意が含まれることもあります。 （注） EULA に同意しない場合は、変更した設定はリロード後に反映されません。デフォルトのライセンス（または非アクティブ化されたライセンス）がリロード後にアクティブになります。
ステップ 2	**license right-to-use** [**activate** | **deactivate**] **addon** [**dna-essentials** | **dna-advantage**] [**all** | **evaluation** | **subscription** {**all** | **slot** *slot-number <1-8>*}] [**acceptEULA**] 例： `Device# license right-to-use activate addon dna-essentials subscription all acceptEULA`	ライセンス レベルをアクティブにします。すべてのスイッチ上でアクティブ化され、EULA への同意が含まれることもあります。 （注） EULA に同意しない場合は、変更した設定はリロード後に反映されません。デフォルトのライセンス（または非アクティブ化されたライセンス）がリロード後にアクティブになります。

ステップ 3	**reload** [*LINE* | **at** | **cancel** | **in** | **slot** *device-member-number* | **standby-cpu**]
	例：
	`Device# reload slot 1`
	`Proceed with reload? [confirm] y`
		特定のデバイスメンバをリロードして、アクティブ化プロセスを完了します。
		ライセンス レベルを変更する場合は、設定を保存する必要はありません。ただし、リロードする前にすべての設定が適切に保存されていることを確認すること

	コマンドまたはアクション	目的
		をお勧めします。再起動時に高いライセンス レベルから低いライセンス レベルに変更すると、適用できない CLI は削除されます。アクティブに使用される低いライセンス レベルの機能はすべて削除されないようにしてください。
ステップ 4	**show license right-to-use usage** [**slot** *slot-number*] **例：** Device# **show license right-to-use usage** Slot# License Name Type usage-duration(y:m:d) In-Use EULA --- 1 network-essentials Permanent 0 :0 :3 no yes 1 network-essentials Evaluation 0 :0 :0 no no 1 network-essentials Subscription 0 :0 :0 no no 1 network-advantage Permanent 0 :0 :6 yes yes 1 network-advantage Evaluation 0 :0 :0 no no 1 network-advantage Subscription 0 :0 :0 no no 1 dna-essentials Evaluation 0 :0 :0 no no 1 dna-essentials Subscription 0 :0 :0 no no 1 dna-advantage Evaluation 0 :0 :0 no no 1 dna-advantage Subscription 0 :0 :6 yes yes --- Device#	詳細な使用状況に関する情報を表示します。

ライセンスの再ホスト

ライセンスを再ホストするには、1 つのデバイスのライセンスを非アクティブ化し、別のデバイスで同じライセンスをアクティブ化します。

手順

	コマンドまたはアクション	目的
ステップ 1	**license right-to-use deactivate [license-level] slot** *slot-num* 例： `Device# license right-to-use deactivate network-essentials slot 1`	1 つのデバイスのライセンスを非アクティブ化します。この例では、Network Essentials ライセンスを検討しています。
ステップ 2	**license right-to-use activate [license-level] slot** *slot-num* [**acceptEULA**] 例： `Device# license right-to-use activate network-essentials slot 2 acceptEULA`	別のデバイスのライセンスをアクティブ化します。この例では、Network Essentials ライセンスを検討しています。

Network Essentials ライセンスから Network Advantage へのアップグレード

手順

	コマンドまたはアクション	目的
ステップ 1	**license right-to-use activate network-advantage slot** *switch-ID***acceptEULA**	Network Advantage ライセンスを有効にします。 スイッチ ID を入力します。 acceptEULA を入力して同意したことを示します。
ステップ 2	**show license right-to-usesummary**	リブート ライセンス レベルが Network Advantage であることを確認します。
ステップ 3	**reload**	スイッチをリブートして Network Advantage でブートします。

Network Essentials ライセンスがある SKU での DNA Essentials ライセンスの有効化

手順

	コマンドまたはアクション	目的
ステップ 1	**license right-to-use activate addon dna-essentials subscriptionslot** *switch-ID***acceptEULA**	DNA Essentials ライセンスを有効にします。 スイッチ ID を入力します。 acceptEULA を入力して同意したことを示します。
ステップ 2	**show license right-to-usesummary**	リブート ライセンス レベルが DNA Essentials であることを確認します。
ステップ 3	**reload**	スイッチをリブートして DNA Essentials でブートします。

Network Essentials ライセンスがある SKU での Network Advantage ライセンスの評価

手順

	コマンドまたはアクション	目的
ステップ 1	**license right-to-use activate network-advantage evaluationslot** *switch-ID***acceptEULA**	Network Advantage 評価ライセンスを有効にします。 スイッチ ID を入力します。 acceptEULA を入力して同意したことを示します。
ステップ 2	**show license right-to-usesummary**	リブート ライセンス レベルが Network Advantage Evaluation であることを確認します。
ステップ 3	**reload**	スイッチをリブートして Network Advantage 評価ライセンスでブートします。

Network Essentials SKU での Evaluation Network Advantage ライセンスの無効化

手順

	コマンドまたはアクション	目的
ステップ 1	**license right-to-use deactivate network-advantage evaluationslot** *switch-ID***acceptEULA**	Network Advantage 評価ライセンスを無効にします。 スイッチ ID を入力します。
ステップ 2	**show license right-to-usesummary**	リブート ライセンス レベルが Network Essentials であることを確認します。
ステップ 3	**reload**	スイッチをリブートして Network Essentials でブートします。

許可されるライセンスの組み合わせの CLI

表 *7*: 基本ライセンス

Network Essentials
license right-to-use [activate | deactivate] network-essentials [acceptEULA] license right-to-use [activate | deactivate] network-essentials evaluation [acceptEULA]
Network Advantage
license right-to-use [activate | deactivate] network-advantage [acceptEULA] license right-to-use [activate | deactivate] network-advantage evaluation [acceptEULA]

表 *8*: アドオン ライセンス

DNA Essentials
license right-to-use [activate | deactivate] addon dna-essentials subscription [acceptEULA] license right-to-use [activate | deactivate] addon dna-essentials evaluation [acceptEULA]
DNA Advantage
license right-to-use [activate | deactivate] addon dna-advantage subscription [acceptEULA] license right-to-use [activate | deactivate] addon dna-advantage evaluation [acceptEULA]

RTU ライセンスのモニタリングおよびメンテナンス

コマンド	目的
show license right-to-use default	デフォルトのライセンス情報を表示します。
show license right-to-use detail	デバイスのライセンスに関する詳細な情報を表示します。
show license right-to-use eula {evaluation | permanent | subscription}	エンドユーザ ライセンス契約を表示します。
show license right-to-use mismatch	一致しないライセンス情報を表示します。
show license right-to-use summary	デバイスのライセンス情報のサマリーを表示します。

RTU ライセンスの設定例

例：RTU ライセンス情報の表示

次に、デバイスの RTU ライセンス情報の例を示します。

基本（永久）ライセンスでの出力例

```
Switch# show license right-to-use summary

      License Name           Type    Period left
-----------------------------------------------
network-essentials      Permanent       Lifetime
-----------------------------------------------


License Level In Use: network-essentials
License Level on Reboot: network-essentials
```

アドオン（期間）ライセンスでの出力例

```
Switch# show license right-to-use summary

Switch#show license right-to-use summary
      License Name           Type    Period left
-----------------------------------------------
    dna-essentials   Subscription   CSSM Managed
     dna-advantage   Subscription   CSSM Managed
-----------------------------------------------


License Level In Use: network-advantage Subscription+dna-advantage Subscription
```

```
License Level on Reboot: network-advantage Subscription+dna-advantage Subscription
```

評価ライセンスでの出力例

```
Switch# show license right-to-use summary

Switch#show license right-to-use summary
      License Name           Type    Period left
-----------------------------------------------
 network-advantage     Evaluation             90
     dna-advantage     Evaluation             90
-----------------------------------------------


License Level In Use: network-advantage Evaluation+dna-advantage Evaluation
License Level on Reboot: network-advantage Evaluation+dna-advantage Evaluation
```

RTU ライセンスに関する追加情報

関連資料

関連項目	参照先
この章で使用するコマンドの完全な構文および使用方法の詳細。	『System Management Command Reference』

標準および RFC

標準/RFC	タイトル
なし	—

MIB

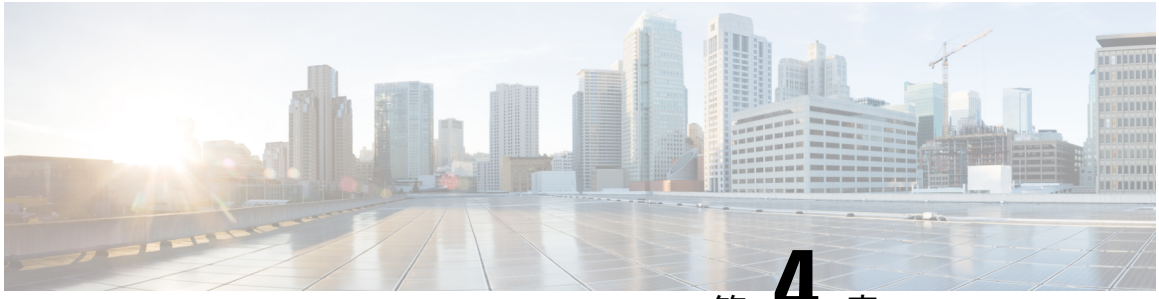
MIB	MIB リンク
オブジェクト ciscoLicenseMIB OID 1.3.6.1.4.1.9.9.359 MIB CISCO-LICENSE-MIB：サポート画像の表示	選択したプラットフォーム、Cisco IOS リリース、およびフィーチャセットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

テクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Product Alert Tool（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

RTU ライセンスの機能履歴と情報

リリース	機能情報
Cisco IOS XE 16.5.1a	この機能が導入されます。



第 4 章

有線ネットワークでの Application Visibility and Control

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。最新の機能情報および警告については、使用するプラットフォームおよびソフトウェア リリースの Bug Search Tool およびリリース ノートを参照してください。このモジュールに記載されている機能の詳細を検索し、各機能がサポートされているリリースのリストを確認する場合は、このモジュールの最後にある機能情報の表を参照してください。

プラットフォームのサポートおよびCisco ソフトウェア イメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator には、http://www.cisco.com/go/cfn からアクセスします。Cisco.com のアカウントは必要ありません。

有線ネットワークでの Application Visibility and Control について

Application Visibility and Control（AVC）は、アプリケーションへの適応力やアプリケーションへのインテリジェンス性に基づいて、厳密なパケットおよび接続からブランチおよびキャンパス ソリューションを発展させるためのシスコの取り組みの重要な部分です。Application Visibility and Control（AVC）は、ネットワークベースのアプリケーション認識（NBAR2）エンジンによるディープ パケット インスペクション技術を使用してアプリケーションを分類します。AVC は、スタンドアロン スイッチおよびスイッチ スタックの有線アクセス ポート上に設定できます。NBAR2 は、プロトコル検出を有効にすることによって明示的に、または **match protocol** 分類子を含む QoS ポリシーを接続することによって暗黙的に、インターフェイス上でアクティブにできます。有線 AVC Flexible Netflow（FNF）をインターフェイス上に設定し、インターフェイスごとのクライアント、サーバ、アプリケーションの統計情報を提供できます。このレコードは、Easy Performance Monitor（Easy perf-mon または ezPM）の **application-statistics** および **application-performance** プロファイルで利用できる **application-client-server-stats** トラフィック監視と同様です。

サポートされる AVC クラス マップおよびポリシー マップのフォーマット

サポートされる **AVC** クラス マップのフォーマット

クラスマップのフォーマット	クラスマップの例	方向
match protocol プロトコル名	`class-map match-any NBAR-VOICE` `match protocol ms-lync-audio`	入力と出力の両方

組み合わせフィルタ	`class-map match-any NBAR-VOICE`
	`match protocol ms-lync-audio`
	`match dscp ef`
		入力と出力の両方

サポートされる **AVC** ポリシーのフォーマット

ポリシーのフォーマット	**QoS** 処理
match protocol フィルタに基づく出力ポリシー	マークおよびポリシー
match protocol フィルタに基づく入力ポリシー	マークおよびポリシー

次の表で、AVC ポリシーの詳細なフォーマット、および例について説明します。

AVC ポリシーのフォーマット	AVC ポリシーの例	方向
ベーシック セット	policy-map MARKING-IN class NBAR-MM_CONFERENCING set dscp af41	入力および出力
ベーシック ポリシー	policy-map POLICING-IN class NBAR-MM_CONFERENCING police cir 600000 set dscp af41	入力および出力
ベーシック セットおよびポリシー	policy-map webex-policy class webex-class set dscp ef cos police 5000000	入力および出力
デフォルトを含む複数のセットおよびポリシー	policy-map webex-policy class webex-class set dscp af31 cos police 4000000 class class-webex-category set dscp ef cos police 6000000 class class-default set dscp <>	入力および出力
階層型ポリシー	policy-map webex-policy class webex-class police 5000000 service-policy client-in-police-only policy-map client-in-police-only class webex-class police 100000 class class-webex-category set dscp ef cos police 200000	入力および出力
階層型セットおよびポリシー	policy-map webex-policy class class-default police 1500000 service policy client-up-child policy-map webex-policy class webex-class police 100000 set dscp ef class class-webex-category police 200000 set dscp af31	

有線 Application Visibility and Control の制限

- NBAR 対応 QoS ポリシー設定は有線物理ポートでのみ許可されます。ポリシー設定は、たとえば、VLAN、ポート チャネル、および他の論理インターフェイスなどの仮想インターフェイスではサポートされていません。
- NBAR2 ベースの一致基準 **match protocol** は、マーキング アクションおよびポリシング アクションでのみ許可されます。NBAR2 一致基準は、キューイング機能が設定されているポリシーでは許可されません。
- 「一致プロトコル」：すべてのポリシーで最大 255 の同時に異なるプロトコル（8 ビットの HW 制限）。
- NBAR2 属性ベースの QoS はサポートされていません（**match protocol** 属性）。
- AVC は管理ポート（Gig 0/0）ではサポートされていません。
- IPv6 パケットの分類はサポートされていません。
- IPv4 ユニキャスト（TCP/UDP）のみがサポートされます。
- Web UI：Web UI からアプリケーションの可視性を設定し、アプリケーションのモニタリングを実行できます。アプリケーション制御は、CLI を使用してのみ実行できます。Web UI ではサポートされていません。

  Web UI 上で有線 AVC のトラフィックを管理、またはチェックするには、最初に CLI を使用して **ip http authentication local** と **ip nbar http-service** コマンドを設定する必要があります。
- NBAR および ACL のロギングは、同一スイッチ上で一緒に設定することはできません。
- 有線 AVC は LAN Base ライセンスではサポートされません。
- プロトコル検出、アプリケーション ベースの QoS、および有線 AVC FNF は、非アプリケーション ベース FNF がある同一インターフェイス上で同時に設定することはできません。ただし、これらの有線 AVC 機能は、相互に設定できます。たとえば、プロトコル検出、アプリケーション ベースの QoS、および有線 AVC FNF は、同一インターフェイス上で同時に設定できます。
- 単一の事前定義されたレコードは、有線 AVC FNF でサポートされています。
- 接続は、物理 Layer2（アクセス/トランク）および Layer3 ポートでのみ行う必要があります。アップリンクは、単一のアップリンクであり、ポートチャネルの一部でなければ接続できます。
- パフォーマンス：各スイッチ メンバーは、50% 未満の CPU 使用率で、1 秒あたり 500 の接続（CPS）を処理できます。

- 拡張性：48 個のアクセス ポートごとに最大 10,000 の双方向フローと、24 個のアクセス ポートごとに 5000 の双方向フローを処理できます。（アクセス ポートごとに～ 200 フロー）。

Application Visibility and Control の設定方法

有線ネットワークでの Application Visibility and Control の設定

有線ポートで Application Visibility and Control を設定するには、次の手順を実行します。

可視性の設定

- インターフェイス コンフィギュレーション モードで **ip nbar protocol-discovery** コマンドを使用してインターフェイス上でプロトコル検出を有効にすることで、NBAR2 エンジンをアクティブ化します。インターフェイスでのアプリケーション認識の有効化（101 ページ）を参照してください。

制御設定：次の手順に従って、アプリケーションに基づいて QoS ポリシーを設定します。

1. AVC QoS ポリシーの作成。AVC QoS ポリシーの作成（102 ページ）を参照してください。
2. インターフェイスへの AVC QoS ポリシーの適用。スイッチ ポートへの QoS ポリシーの適用（105 ページ）を参照してください。

アプリケーション ベースの Flexible Netflow の設定：

- フローにキー フィールドおよび非キー フィールドを指定して、フロー レコードを作成します。フロー レコードの作成（106 ページ）を参照してください。
- フローエクスポータを作成してフロー レコードをエクスポートします。フローエクスポータの作成（109 ページ）を参照してください。
- フロー レコードおよびフロー エクスポータに基づいて、フロー モニタを作成します。フロー モニタの作成（110 ページ）を参照してください。
- インターフェイスにフロー モニタを接続します。インターフェイスへのフロー モニタの関連付け（112 ページ）を参照してください。

プロトコル検出、アプリケーション ベースの QoS およびアプリケーション ベースの FNF は、すべて独立した機能です。単独で設定することも、または同じインターフェイスで同時に設定することもできます。

インターフェイスでのアプリケーション認識の有効化

インターフェイス上でアプリケーション認識をイネーブルにするには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 例： `Device# configure terminal`	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** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	プロトコル検出をイネーブルにするインターフェイスを指定し、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	**ip nbar protocol-discovery** 例： `Device(config-if)# ip nbar protocol-discovery`	NBAR2 エンジンをアクティブ化することで、インターフェイスでアプリケーション認識を有効にします。
ステップ 4	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。

AVC QoS ポリシーの作成

AVC QoS ポリシーを作成するには、次の一般的な手順を実行します。

1. match protocol フィルタでクラス マップを作成します。
2. ポリシー マップを作成します。
3. インターフェイスにポリシー マップを適用します。

クラス マップの作成

match protocol フィルタを設定する前に、クラス マップを作成する必要があります。マーキングやポリシングなどの QoS アクションをトラフィックに適用できます。AVC の match protocol フィルタは、有線アクセス ポートに適用されます。サポートされているプロトコルの詳細については、http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/qos_nbar/prot_lib/config_library/nbar-prot-pack-library.html を参照してください。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	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	**class-map** *class-map-name* 例： `Device(config)# class-map webex-class`	クラス マップを作成します。

ステップ 3	**match protocol** *application-name*
	例：
	`Device(config)# class-map webex-class`
	`Device(config-cmap)# match protocol webex-media`
		アプリケーション名との一致を指定します。

ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。また、Ctrl+Z キーを押しても、グローバル コンフィギュレーション モードを終了できます。

ポリシー マップの作成

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	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	**policy-map** *policy-map-name* 例： `Device(config)# policy-map webex-policy`	ポリシーマップ名を入力することによってポリシー マップを作成し、ポリシー マップ コンフィギュレーション モードを開始します。 デフォルトでは、ポリシー マップは定義されていません。 ポリシー マップのデフォルトの動作では、パケットが IP パケットの場合は DSCP が 0 に、パケットがタグ付きの場合は CoS が 0 に設定されます。ポリシングは実行されません。

	コマンドまたはアクション	目的
		（注） 既存のポリシー マップを削除するには、**no policy-map** *policy-map-name* グローバル コンフィギュレーション コマンドを使用します。
ステップ 3	**class** [*class-map-name* | **class-default**] 例： `Device(config-pmap)# class webex-class`	トラフィックの分類を定義し、ポリシー マップ クラス コンフィギュレーション モードを開始します。 デフォルトでは、ポリシー マップおよびクラスマップは定義されていません。 すでに **class-map** グローバル コンフィギュレーション コマンドを使用してトラフィック クラスが定義されている場合は、このコマンドで *class-map-name* にその名前を指定します。 **class-default** トラフィック クラスは定義済みで、どのポリシーにも追加できます。このトラフィック クラスは、常にポリシーマップの最後に配置されます。暗黙の **match any** が **class-default** クラスに含まれている場合、他のトラフィック クラスと一致していないすべてのパケットは **class-default** と一致します。 （注） 既存のクラス マップを削除するには、**no class** *class-map-name* ポリシー マップ コンフィギュレーション コマンドを使用します。
ステップ 4	**police** *rate-bps burst-byte* 例： `Device(config-pmap-c)# police 100000 80000`	分類したトラフィックにポリサーを定義します。 デフォルトでは、ポリサーは定義されていません。 • *rate-bps* には、平均トラフィック レートをビット/秒（bps）で指定します。指定できる範囲は 8000 ～ 10000000000 です。

	コマンドまたはアクション	目的
		• *burst-byte* には、標準バースト サイズをバイト数で指定します。指定できる範囲は 8000 ～ 1000000 です。
ステップ 5	**set** { **dscp** *new-dscp* | **cos** *cos-value* } 例： `Device(config-pmap-c)# set dscp 45`	パケットに新しい値を設定することによって、IP トラフィックを分類します。 • **dscp** *new-dscp* には、分類されたトラフィックに割り当てる新しい DSCP 値を入力します。指定できる範囲は 0 ～ 63 です。
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。また、Ctrl+Z キーを押しても、グローバル コンフィギュレーション モードを終了できます。

スイッチ ポートへの QoS ポリシーの適用

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	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** *interface-id* 例： `Device(config)# interface Gigabitethernet 1/0/1`	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	**service-policy input** *policymapname* 例： `Device(config-if)# service-policy input MARKING_IN`	インターフェイスにローカル ポリシーを適用します。
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。また、Ctrl+Z キーを押しても、グローバル コンフィギュレーション モードを終了できます。

有線 AVC Flexible Netflow の設定

フロー レコードの作成

1 つのフロー レコードを設定して、フロー モニタに関連付けることができます。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	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	**flowrecord** *flow_record_name* 例： `Device(config)# flow record flow-record-1`	フロー レコード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	**description** 説明 例： `Device(config-flow-record)# description flow-record-1`	（任意）フロー レコードの説明を作成します。
ステップ 4	**matchipv4version** 例： `Device (config-flow-record)# match ipv4 version`	IPv4 ヘッダーからの IP バージョンとの一致を指定します。
ステップ 5	**matchipv4protocol** 例： `Device (config-flow-record)# match ipv4 protocol`	IPv4 プロトコルとの一致を指定します。
ステップ 6	**matchapplicationname** 例： `Device (config-flow-record)# match application name`	アプリケーション名との一致を指定します。 （注） この操作は、AVC サポートでは必須です。フローがアプリケーションと一致することが可能になるためです。
ステップ 7	**match connection client ipv4 address** 例： `Device (config-flow-record)# match connection client ipv4 address`	クライアント（フローイニシエータ）の IPv4 アドレスとの一致を指定します。

	コマンドまたはアクション	目的
ステップ 8	**match connection server ipv4 address** 例： `Device (config-flow-record)# match connection server ipv4 address`	サーバ（フロー レスポンダ）の IPv4 アドレスとの一致を指定します。
ステップ 9	**match connection server transport port** 例： `Device (config-flow-record)# match connection server transport port`	サーバのトランスポートポートとの一致を指定します。
ステップ 10	**match flow observation point** 例： `Device (config-flow-record)# match flow observation point`	フロー観測メトリックの観測ポイント ID との一致を指定します。
ステップ 11	**collect flow direction** 例： `Device (config-flow-record)# collect flow direction`	次の手順で **collect connection initiator** コマンドの **initiator** キーワードで指定される双方向フローの関連する側（イニシエータまたはレスポンダ）の方向（入力または出力）を収集するように指定します。**initiator** キーワードで指定される値に応じて、**flow direction** キーワードは次の値をとります。 • 0x01 = 入力フロー • 0x02 = 出力フロー **initiator** キーワードがイニシエータに設定されている場合、フローの方向はフローのイニシエータ側から指定されます。initiator キーワードがレスポンダに設定されている場合、フローの方向はフローのレスポンダ側から指定されます。有線 AVC では、**initiator** キーワードは常にイニシエータに設定されています。
ステップ 12	**collect connection initiator** 例： `Device (config-flow-record)# collect connection initiator`	**collect flow direction** コマンドで指定されたフローの方向に関連するフローの側（イニシエータまたはレスポンダ）を収集するように指定します。**initiator** キーワードは、フローの方向に関する次の情報を提供します。

	コマンドまたはアクション	目的
		• 0x01 = イニシエータ：フローの送信元は接続のイニシエータです 有線 AVC では、**initiator** キーワードは常にイニシエータに設定されています。
ステップ 13	**collect connection client counter packets long** 例： `Device (config-flow-record)# collect connection client counter packets long`	クライアントが送信したパケット数を収集するように指定します。
ステップ 14	**collect connection client counter bytes network long** 例： `Device (config-flow-record)# collect connection client counter bytes network long`	クライアントが送信したバイト数の合計を収集するように指定します。
ステップ 15	**collect connection server counter packets long** 例： `Device (config-flow-record)# collect connection server counter packets long`	サーバが送信したパケット数を収集するように指定します。
ステップ 16	**collect connection server counter bytes network long** 例： `Device (config-flow-record)# collect connection server counter bytes network long`	サーバが送信したバイト数の合計を収集するように指定します。
ステップ 17	**collect timestamp absolute first** 例： `Device (config-flow-record)# collect timestamp absolute first`	最初のパケットがフローで確認されたときの時間をミリ秒単位で収集するように指定します。
ステップ 18	**collect timestamp absolute last** 例： `Device (config-flow-record)# collect timestamp absolute last`	最新のパケットがフローで確認されたときの時間をミリ秒単位で収集するように指定します。
ステップ 19	**collect connection new-connections** 例：	観測された接続開始の数を収集するように指定します。

	コマンドまたはアクション	目的
	`Device (config-flow-record)# collect connection new-connections`	
ステップ 20	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。また、Ctrl+Z キーを押しても、グローバル コンフィギュレーション モードを終了できます。
ステップ 21	**show flow record** 例： `Device # show flow record`	すべてのフロー レコードに関する情報を表示します。

フロー エクスポータの作成

フロー エクスポータを作成すると、フローのエクスポート パラメータを定義できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	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	**flowexporter** *flow_exporter_name* 例： `Device(config)# flow exporter flow-exporter-1`	フロー エクスポータ 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	**description** 説明 例： `Device(config-flow-exporter)# description flow-exporter-1`	（任意）フロー エクスポータの説明を作成します。
ステップ 4	**destination** { *hostname* | *ipv4-address* | *ipv6-address* } 例： `Device (config-flow-exporter)# destination 10.10.1.1`	エクスポータでデータを送信する宛先システムのホスト名、IPv4 または IPv6 アドレスを指定します。
ステップ 5	**option application-table** [**timeout** 秒] 例： `Device (config-flow-exporter)# option application-table timeout 500`	（任意）フロー エクスポータのアプリケーション テーブルのオプションを設定します。**timeout** オプションを使用すると、フロー エクスポータの再送信時間を秒単位で設定できます。有効な範囲は 1 ～ 86400 秒です。

	コマンドまたはアクション	目的
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。また、Ctrl+Z キーを押しても、グローバル コンフィギュレーション モードを終了できます。
ステップ 7	**show flow exporter** 例： `Device # show flow exporter`	すべてのフロー エクスポータに関する情報を表示します。
ステップ 8	**show flow exporter statistics** 例： `Device # show flow exporter statistics`	フロー エクスポータの統計情報を表示します。

フロー モニタの作成

フロー モニタを作成して、フロー レコードに関連付けることができます。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	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	**flow monitor** *monitor-name* 例： `Device (config)# flow monitor flow-monitor-1`	フローモニタを作成し、フローモニタ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	**description** 説明 例： `Device (config-flow-monitor)# description flow-monitor-1`	（任意）フローモニタの説明を作成します。
ステップ 4	**record** *record-name* 例： `Device (config-flow-monitor)# record flow-record-1`	事前に作成されたレコードの名前を指定します。
ステップ 5	**exporter** *exporter-name* 例： `Device (config-flow-monitor)# exporter flow-exporter-1`	事前に作成されたエクスポータ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 6	**cache** { **entries** *number-of-entries* | **timeout** {**active** | **inactive**} | **type normal** }
	例：
	`Device (config-flow-monitor)# cache timeout active 1800`
	例：
	`Device (config-flow-monitor)# cache timeout inactive 200`
	例：
	`Device (config-flow-monitor)# cache type normal`
		（任意）フローキャッシュパラメータを設定するように指定します。
		• **entries** *number-of-entries*：フローキャッシュ内のフローエントリの最大数を 16 ～ 65536 の範囲で指定します。
		（注） 標準のキャッシュタイプのみがサポートされます。

ステップ 7	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。また、Ctrl+Z キーを押しても、グローバル コンフィギュレーションモードを終了できます。
ステップ 8	**show flow monitor** 例： `Device # show flow monitor`	すべてのフローモニタに関する情報を表示します。
ステップ 9	**show flow monitor wdavc** 例： `Device # show flow monitor wdavc`	指定した有線 AVC フロー モニタに関する情報を表示します。
ステップ 10	**show flow monitor wdavc statistics** 例： `Device# show flow monitor wdavc statistics`	有線 AVC フロー モニタの統計情報を表示します。
ステップ 11	**clear flow monitor wdavc statistics** 例： `Device# clear flow monitor wdavc statistics`	指定したフローモニタの統計情報をクリアします。**clear flow monitor wdavc statistics** の後に **show flow monitor wdavc statistics** コマンドを使用して、すべての統計情報がリセットされていることを確認します。
ステップ 12	**show flow monitor wdavc cache format table** 例： `Device# show flow monitor wdavc cache format table`	表形式でフローキャッシュの内容を表示します。

	コマンドまたはアクション	目的
ステップ 13	**show flow monitor wdavc cache format record** 例： `Device# show flow monitor wdavc cache format record`	フローレコードと同様の形式でフローキャッシュの内容を表示します。
ステップ 14	**show flow monitor wdavc cache format csv** 例： `Device# show flow monitor wdavc cache format csv`	CSV 形式でフローキャッシュの内容を表示します。

インターフェイスへのフロー モニタの関連付け

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	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** *interface-id* 例： `Device(config)# interface Gigabitethernet 1/0/1`	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	**ip flow monitor** *monitor-name*{ **input** | **output** } 例： `Device (config-if) # ip flow monitor flow-monitor-1 input`	入力パケットと出力パケットの両方またはいずれか用のインターフェイスにフロー モニタを関連付けます。
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。また、Ctrl+Z キーを押しても、グローバル コンフィギュレーション モードを終了できます。

NBAR2 カスタム アプリケーション

NBAR2 では、カスタム プロトコルを使用してカスタム アプリケーションを識別できます。カスタム プロトコルは、プロトコルとアプリケーションをサポートしますが、現在のところ、NBAR2 はサポートしていません。

すべての展開において、シスコが提供する NBAR2 プロトコル パックの対象外であるローカル アプリケーションおよび特定のアプリケーションがあります。ローカルアプリケーションは主に次のように分類されます。

- 組織への特定のアプリケーション
- 地域特有のアプリケーション

NBAR2 では、このようなローカル アプリケーションを手動でカスタマイズする方法を提供しています。グローバル コンフィギュレーション モードで **ip nbar custom** *myappname* コマンドを使用して、手動でアプリケーションをカスタマイズできます。カスタム アプリケーションは、組み込みプロトコルより優先されます。それぞれのカスタムプロトコルでは、ユーザは、レポート目的に使用できるセレクタ ID を定義できます。

さまざまなタイプのアプリケーション カスタマイズがあります。

一般的なプロトコルのカスタマイズ

- HTTP
- SSL
- DNS

コンポジット：複数の基本的なプロトコルに基づくカスタマイズ： **server-name**

レイヤ 3/レイヤ 4 のカスタマイズ

- IPv4 アドレス（IPv4 address）
- DSCP 値
- TCP/UDP ポート
- フロー送信元または宛先の方向

バイト オフセット：ペイロードの特定のバイト値に基づくカスタマイズ

HTTP のカスタマイズ

HTTP のカスタマイズは、次の HTTP フィールドの組み合わせに基づいて実行できます。

- **cookie** ：HTTP クッキー
- **host** ：リソースを含む元のサーバのホスト名
- **method** ：HTTP メソッド
- **referrer** ：リソース リクエストの取得元のアドレス
- **url** ：Uniform Resource Locator のパス
- **user-agent** ：要求を送信するエージェントによって使用されているソフトウェア
- **version** ：HTTP バージョン

- **via** ：HTTP 経由フィールド

HTTP のカスタマイズ

セレクタ ID 10 が付いた HTTP ホスト「*mydomain.com」を使用する MYHTTP と呼ばれるカスタム アプリケーション。

```
Device# configure terminal
Device(config)# ip nbar custom MYHTTP http host *mydomain.com id 10
```

SSL のカスタマイズ

SSL サーバ名指定（SNI）または共通名（CN）から抽出した情報を使用して、SSL 暗号化トラフィックでカスタマイズを行うことができます。

SSL のカスタマイズ

セレクタ ID 11 が付いた SSL 固有名「mydomain.com」を使用する MYSSL と呼ばれるカスタム アプリケーション。

```
Device# configure terminal
Device(config)#ip nbar custom MYSSL ssl unique-name *mydomain.com id 11
```

DNS のカスタマイズ

NBAR2 は、DNS 要求および応答トラフィックを確認し、アプリケーションへの DNS 応答に関連付けることができます。DNS 応答から戻された IP アドレスはキャッシュされ、その特定のアプリケーションに関連付けられているその後のパケット フローに使用されます。

ip nbar custom *application-name* **dns** *domain-name* **id** *application-id* コマンドは、DNS のカスタマイズに使用されます。既存のアプリケーションを拡張するには、**ip nbar custom** *application-name* **dns domain-name** *domain-name* **extends** *existing-application* コマンドを使用します。

DNS ベースのカスタマイズの詳細については、http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/qos_nbar/configuration/xe-3s/asr1000/qos-nbar-xe-3s-asr-1000-book/nbar-custapp-dns-xe.html を参照してください。

DNS のカスタマイズ

セレクタ ID 12 が付いた DNS ドメイン名「mydomain.com」を使用する MYDNS と呼ばれるカスタム アプリケーション。

```
Device# configure terminal
Device(config)# ip nbar custom MYDNS dns domain-name *mydomain.com id 12
```

複合カスタマイズ

NBAR2 では、HTTP、SSL または DNS に現れるドメイン名に基づいてアプリケーションをカスタマイズする方法が提供されます。

複合カスタマイズ

セレクタ ID 13 が付いた HTTP、SSL または DNS ドメイン名「mydomain.com」を使用する MYDOMAIN と呼ばれるカスタム アプリケーション。

```
Device# configure terminal
Device(config)# ip nbar custom MYDOMAIN composite server-name *mydomain.com id 13
```

L3/L4 のカスタマイズ

レイヤ3/レイヤ4のカスタマイズは、パケットタプルに基づいており、フローの最初のパケットで常に一致します。

L3/L4 のカスタマイズ

IP アドレス 10.56.1.10 および 10.56.1.11、セレクタ ID 14 が付いた TCP および DSCP ef に一致する LAYER4CUSTOM と呼ばれるカスタム アプリケーション。

```
Device# configure terminal
Device(config)# ip nbar custom LAYER4CUSTOM transport tcp id 14
Device(config-custom)# ip address 10.56.1.10 10.56.1.11
Device(config-custom)# dscp ef
```

例：カスタム アプリケーションのモニタリング

カスタム アプリケーションのモニタリングのための show コマンド

show ip nbar protocol-id | inc Custom

```
Device# show ip nbar  protocol-id | inc Custom
LAYER4CUSTOM              14             Custom
MYDNS                     12             Custom
MYDOMAIN                  13             Custom
MYHTTP                    10             Custom
MYSSL                     11             Custom
```

show ip nbar protocol-discovery protocol *CUSTOM_APP*

```
WSW-157# show ip nbar  protocol-id MYSSL
Protocol Name              id             type
-----------------------------------------------
MYSSL                     11             Custom
```

NBAR2 ダイナミック ヒットレス プロトコル パックのアップグレード

プロトコル パックは、デバイスのシスコ ソフトウェアを置き換えることなく、デバイスの NBAR2 プロトコル サポートを更新するソフトウェア パッケージです。プロトコル パックには、NBAR2 によって正式にサポートされている、コンパイル済みでパック済みのアプリケーションに関する情報が含まれています。各アプリケーションについて、プロトコル パックには、アプリケーション署名とアプリケーション属性の情報が含まれています。各ソフトウェア リリースには、組み込みのプロトコル パックがバンドルされています。

プロトコル パックには次の特長があります。

- ロードが容易で高速。
- 高いバージョンのプロトコルパックにアップグレードしたり、低いバージョンのプロトコル パックに戻したりするのが容易。
- スイッチのリロードを必要としない。

NBAR2 プロトコル パックは、次の URL から Cisco Software Center でダウンロードできます：https://software.cisco.com/download/navigator.html

Application Visibility and Control のモニタリング

このセクションでは、アプリケーションの可視性に関する新しいコマンドについて説明します。

次のコマンドは、およびアクセスポートのアプリケーションの可視性をモニタするために使用できます。

表 *9*: のアプリケーションの可視性モニタリング コマンド

コマンド	目的

例：Application Visibility and Control の設定

次に、match protocol でアプリケーション名のフィルタを適用してクラス マップを作成する例を示します。

```
Device# configure terminal
Device(config)# class-map match-any NBAR-VOICE
Device(config-cmap)# match protocol ms-lync-audio
Device(config-cmap)#end
```

次に、ポリシー マップを作成し、出力 QoS の既存のクラス マップを定義する例を示します。

```
Device# configure terminal
Device(config)# policy-map test-avc-up
Device(config-pmap)# class cat-browsing
Device(config-pmap-c)# police 150000
Device(config-pmap-c)# set dscp 12
Device(config-pmap-c)#end
```

次に、ポリシー マップを作成し、入力 QoS の既存のクラス マップを定義する例を示します。

```
Device# configure terminal
Device(config)# policy-map test-avc-down
Device(config-pmap)# class cat-browsing
Device(config-pmap-c)# police 200000
Device(config-pmap-c)# set dscp 10
Device(config-pmap-c)#end
```

次に、ポリシー マップをスイッチ ポートに適用する例を示します。

```
Device# configure terminal
Device(config)# interface GigabitEthernet 1/0/1
Device(config-if)# switchport mode access
Device(config-if)# switchport access vlan 20
Device(config-if)# service-policy input POLICING_IN
Device(config-if)#end
```

show コマンドによる設定の表示

show ip nbar protocol-discovery

インターフェイスごとのプロトコル検出統計情報のレポートを表示します。

次に、インターフェイスごとの統計情報の出力例を示します。

```
Deviceqos-cat9k-reg2-r1# show ip nbar protocol-discovery int GigabitEthernet1/0/1


GigabitEthernet1/0/1
Last clearing of "show ip nbar protocol-discovery" counters 00:03:16

                                  Input
Output
                                  -----
------
Protocol                          Packet Count
Packet Count
                                  Byte Count
Byte Count
                                  30sec Bit Rate (bps)
30sec Bit Rate (bps)
                                  30sec Max Bit Rate (bps)
30sec Max Bit Rate (bps)
------------------------ ------------------------
-----------------------------------------------------
ms-lync                           60580
55911
                                  31174777
28774864
                                  3613000
93000
                                  3613000
3437000
Total                             60580
55911
                                  31174777
28774864
                                  3613000
93000
                                  3613000
3437000
```

show policy-map interface

すべてのインターフェイス上のQoS統計情報および設定済みのポリシーマップを表示します。

次に、すべてのインターフェイスに設定されたポリシー マップの出力例を示します。

```
Deviceqos-cat9k-reg2-r1# show policy-map int


GigabitEthernet1/0/1
 Service-policy input: MARKING-IN

   Class-map: NBAR-VOICE (match-any)
     718 packets
     Match: protocol ms-lync-audio
       0 packets, 0 bytes
       30 second rate 0 bps
     QoS Set
       dscp ef

   Class-map: NBAR-MM_CONFERENCING (match-any)
     6451 packets
     Match: protocol ms-lync
       0 packets, 0 bytes
       30 second rate 0 bps
     Match: protocol ms-lync-video
       0 packets, 0 bytes
       30 second rate 0 bps
     QoS Set
       dscp af41

   Class-map: class-default (match-any)
     34 packets
     Match: any
```

show コマンドによるフロー モニタ設定の表示

show flow monitor wdavc

指定した有線 AVC フロー モニタに関する情報を表示します。

```
Device # show flow monitor wdavc

Flow Monitor wdavc:
  Description:       User defined
  Flow Record:       wdavc
  Flow Exporter:     wdavc-exp (inactive)
  Cache:
    Type:                normal (Platform cache)
    Status:              not allocated
    Size:                12000 entries
    Inactive Timeout:    15 secs
    Active Timeout:      1800 secs
```

show flow monitor wdavc statistics

有線 AVC フロー モニタの統計情報を表示します。

```
Device# show flow monitor wdavc statistics
  Cache type:                                  Normal (Platform cache)
  Cache size:                                   12000
  Current entries:                                 13

  Flows added:                                     26
  Flows aged:                                      13
    - Active timeout      (  1800 secs)           1
    - Inactive timeout    (    15 secs)          12
```

clear flow monitor wdavc statistics

指定したフロー モニタの統計情報をクリアします。**clear flow monitor wdavc statistics** の後に **show flow monitor wdavc statistics** コマンドを使用して、すべての統計情報がリセットされていることを確認します。以下に、フロー モニタ統計情報をクリアした後の **show flow monitor wdavc statistics** コマンドのサンプル出力を示します。

```
Device# show flow monitor wdavc statistics
  Cache type:                                  Normal (Platform cache)
  Cache size:                                   12000
  Current entries:                                  0

  Flows added:                                      0
  Flows aged:                                       0
```

show コマンドによるキャッシュの内容の表示

show flow monitor wdavc cache format table

表形式でフロー キャッシュの内容を表示します。

```
Device# show flow monitor wdavc cache format table
  Cache type:                                  Normal (Platform cache)
  Cache size:                                   12000
  Current entries:                                 13

  Flows added:                                     26
  Flows aged:                                      13
    - Active timeout      (  1800 secs)           1
    - Inactive timeout    (    15 secs)          12

CONN IPV4 INITIATOR ADDR  CONN IPV4 RESPONDER ADDR  CONN RESPONDER PORT
  FLOW OBSPOINT ID  IP VERSION  IP PROT  APP NAME
flow dirn ..................
------------------------  ------------------------  -------------------
  ----------------  ----------  -------  ---------------------------
---------
64.103.125.147            144.254.71.184
53        4294967305           4       17  port dns
  Input     ...................
64.103.121.103            10.1.1.2
67        4294967305           4       17  layer7 dhcp
  Input     ....contd...........
64.103.125.3              64.103.125.97
```

```
68        4294967305           4       17  layer7 dhcp
  Input     ....................
10.0.2.6                  157.55.40.149                            443
        4294967305           4        6  layer7 ms-lync
Input     ....................
64.103.126.28             66.163.36.139                            443
        4294967305           4        6  layer7 cisco-jabber-im
Input     ....contd...........
64.103.125.2              64.103.125.29
68        4294967305           4       17  layer7 dhcp
  Input     ....................
64.103.125.97             64.103.101.181
67        4294967305           4       17  layer7 dhcp
  Input     ....................
192.168.100.6             10.10.20.1                              5060
        4294967305           4       17  layer7 cisco-jabber-control
Input     ....contd...........
64.103.125.3              64.103.125.29
68        4294967305           4       17  layer7 dhcp
  Input     ....................
10.80.101.18              10.80.101.6                             5060
        4294967305           4        6  layer7 cisco-collab-control
Input     ....................
10.1.11.4                 66.102.11.99
80        4294967305           4        6  layer7 google-services
  Input     ....contd...........
64.103.125.2              64.103.125.97
68        4294967305           4       17  layer7 dhcp
  Input     ....................
64.103.125.29             64.103.101.181
67        4294967305           4       17  layer7 dhcp
  Input     ....................
```

show flow monitor wdavc cache format record

フロー レコードと同様の形式でフロー キャッシュの内容を表示します。

```
Device# show flow monitor wdavc cache format record
  Cache type:                                Normal (Platform cache)
  Cache size:                                 12000
  Current entries:                               13

  Flows added:                                   26
  Flows aged:                                    13
    - Active timeout      (  1800 secs)          1
    - Inactive timeout    (    15 secs)         12

CONNECTION IPV4 INITIATOR ADDRESS:        64.103.125.147
CONNECTION IPV4 RESPONDER ADDRESS:        144.254.71.184
CONNECTION RESPONDER PORT:                53
FLOW OBSPOINT ID:                         4294967305
IP VERSION:                               4
IP PROTOCOL:                              17
```

```
APPLICATION NAME:                          port dns
flow direction:                            Input
timestamp abs first:                       08:55:46.917
timestamp abs last:                        08:55:46.917
connection initiator:                      Initiator
connection count new:                      2
connection server packets counter:         1
connection client packets counter:         1
connection server network bytes counter:   190
connection client network bytes counter:   106

CONNECTION IPV4 INITIATOR ADDRESS:         64.103.121.103
CONNECTION IPV4 RESPONDER ADDRESS:         10.1.1.2
CONNECTION RESPONDER PORT:                 67
FLOW OBSPOINT ID:                          4294967305
IP VERSION:                                4
IP PROTOCOL:                               17
APPLICATION NAME:                          layer7 dhcp
flow direction:                            Input
timestamp abs first:                       08:55:47.917
timestamp abs last:                        08:55:47.917
connection initiator:                      Initiator
connection count new:                      1
connection server packets counter:         0
connection client packets counter:         1
connection server network bytes counter:   0
connection client network bytes counter:   350

CONNECTION IPV4 INITIATOR ADDRESS:         64.103.125.3
CONNECTION IPV4 RESPONDER ADDRESS:         64.103.125.97
CONNECTION RESPONDER PORT:                 68
FLOW OBSPOINT ID:                          4294967305
IP VERSION:                                4
IP PROTOCOL:                               17
APPLICATION NAME:                          layer7 dhcp
flow direction:                            Input
timestamp abs first:                       08:55:47.917
timestamp abs last:                        08:55:53.917
connection initiator:                      Initiator
connection count new:                      1
connection server packets counter:         0
connection client packets counter:         4
connection server network bytes counter:   0
connection client network bytes counter:   1412

CONNECTION IPV4 INITIATOR ADDRESS:         10.0.2.6
CONNECTION IPV4 RESPONDER ADDRESS:         157.55.40.149
CONNECTION RESPONDER PORT:                 443
FLOW OBSPOINT ID:                          4294967305
IP VERSION:                                4
IP PROTOCOL:                               6
```

```
APPLICATION NAME:                          layer7 ms-lync
flow direction:                            Input
timestamp abs first:                       08:55:46.917
timestamp abs last:                        08:55:46.917
connection initiator:                      Initiator
connection count new:                      2
connection server packets counter:         10
connection client packets counter:         14
connection server network bytes counter:   6490
connection client network bytes counter:   1639

CONNECTION IPV4 INITIATOR ADDRESS:         64.103.126.28
CONNECTION IPV4 RESPONDER ADDRESS:         66.163.36.139
CONNECTION RESPONDER PORT:                 443
FLOW OBSPOINT ID:                          4294967305
IP VERSION:                                4
IP PROTOCOL:                               6
APPLICATION NAME:                          layer7 cisco-jabber-im
flow direction:                            Input
timestamp abs first:                       08:55:46.917
timestamp abs last:                        08:55:46.917
connection initiator:                      Initiator
connection count new:                      2
connection server packets counter:         12
connection client packets counter:         10
connection server network bytes counter:   5871
connection client network bytes counter:   2088

CONNECTION IPV4 INITIATOR ADDRESS:         64.103.125.2
CONNECTION IPV4 RESPONDER ADDRESS:         64.103.125.29
CONNECTION RESPONDER PORT:                 68
FLOW OBSPOINT ID:                          4294967305
IP VERSION:                                4
IP PROTOCOL:                               17
APPLICATION NAME:                          layer7 dhcp
flow direction:                            Input
timestamp abs first:                       08:55:47.917
timestamp abs last:                        08:55:47.917
connection initiator:                      Initiator
connection count new:                      1
connection server packets counter:         0
connection client packets counter:         2
connection server network bytes counter:   0
connection client network bytes counter:   712

CONNECTION IPV4 INITIATOR ADDRESS:         64.103.125.97
CONNECTION IPV4 RESPONDER ADDRESS:         64.103.101.181
CONNECTION RESPONDER PORT:                 67
FLOW OBSPOINT ID:                          4294967305
IP VERSION:                                4
IP PROTOCOL:                               17
```

```
APPLICATION NAME:                          layer7 dhcp
flow direction:                            Input
timestamp abs first:                       08:55:47.917
timestamp abs last:                        08:55:47.917
connection initiator:                      Initiator
connection count new:                      1
connection server packets counter:         0
connection client packets counter:         1
connection server network bytes counter:   0
connection client network bytes counter:   350

CONNECTION IPV4 INITIATOR ADDRESS:         192.168.100.6
CONNECTION IPV4 RESPONDER ADDRESS:         10.10.20.1
CONNECTION RESPONDER PORT:                 5060
FLOW OBSPOINT ID:                          4294967305
IP VERSION:                                4
IP PROTOCOL:                               17
APPLICATION NAME:                          layer7 cisco-jabber-control
flow direction:                            Input
timestamp abs first:                       08:55:46.917
timestamp abs last:                        08:55:46.917
connection initiator:                      Initiator
connection count new:                      1
connection server packets counter:         0
connection client packets counter:         2
connection server network bytes counter:   0
connection client network bytes counter:   2046

CONNECTION IPV4 INITIATOR ADDRESS:         64.103.125.3
CONNECTION IPV4 RESPONDER ADDRESS:         64.103.125.29
CONNECTION RESPONDER PORT:                 68
FLOW OBSPOINT ID:                          4294967305
IP VERSION:                                4
IP PROTOCOL:                               17
APPLICATION NAME:                          layer7 dhcp
flow direction:                            Input
timestamp abs first:                       08:55:47.917
timestamp abs last:                        08:55:47.917
connection initiator:                      Initiator
connection count new:                      1
connection server packets counter:         0
connection client packets counter:         2
connection server network bytes counter:   0
connection client network bytes counter:   712

CONNECTION IPV4 INITIATOR ADDRESS:         10.80.101.18
CONNECTION IPV4 RESPONDER ADDRESS:         10.80.101.6
CONNECTION RESPONDER PORT:                 5060
FLOW OBSPOINT ID:                          4294967305
IP VERSION:                                4
IP PROTOCOL:                               6
```

```
APPLICATION NAME:                          layer7 cisco-collab-control
flow direction:                            Input
timestamp abs first:                       08:55:46.917
timestamp abs last:                        08:55:47.917
connection initiator:                      Initiator
connection count new:                      2
connection server packets counter:         23
connection client packets counter:         27
connection server network bytes counter:   12752
connection client network bytes counter:   8773

CONNECTION IPV4 INITIATOR ADDRESS:         10.1.11.4
CONNECTION IPV4 RESPONDER ADDRESS:         66.102.11.99
CONNECTION RESPONDER PORT:                 80
FLOW OBSPOINT ID:                          4294967305
IP VERSION:                                4
IP PROTOCOL:                               6
APPLICATION NAME:                          layer7 google-services
flow direction:                            Input
timestamp abs first:                       08:55:46.917
timestamp abs last:                        08:55:46.917
connection initiator:                      Initiator
connection count new:                      2
connection server packets counter:         3
connection client packets counter:         5
connection server network bytes counter:   1733
connection client network bytes counter:   663

CONNECTION IPV4 INITIATOR ADDRESS:         64.103.125.2
CONNECTION IPV4 RESPONDER ADDRESS:         64.103.125.97
CONNECTION RESPONDER PORT:                 68
FLOW OBSPOINT ID:                          4294967305
IP VERSION:                                4
IP PROTOCOL:                               17
APPLICATION NAME:                          layer7 dhcp
flow direction:                            Input
timestamp abs first:                       08:55:47.917
timestamp abs last:                        08:55:53.917
connection initiator:                      Initiator
connection count new:                      1
connection server packets counter:         0
connection client packets counter:         4
connection server network bytes counter:   0
connection client network bytes counter:   1412

CONNECTION IPV4 INITIATOR ADDRESS:         64.103.125.29
CONNECTION IPV4 RESPONDER ADDRESS:         64.103.101.181
CONNECTION RESPONDER PORT:                 67
FLOW OBSPOINT ID:                          4294967305
IP VERSION:                                4
IP PROTOCOL:                               17
```

```
APPLICATION NAME:                          layer7 dhcp
flow direction:                            Input
timestamp abs first:                       08:55:47.917
timestamp abs last:                        08:55:47.917
connection initiator:                      Initiator
connection count new:                      1
connection server packets counter:         0
connection client packets counter:         1
connection server network bytes counter:   0
connection client network bytes counter:   350
```

show flow monitor wdavc cache format csv

CSV 形式でフロー キャッシュの内容を表示します。

```
Device# show flow monitor wdavc cache format csv
  Cache type:                                 Normal (Platform cache)
  Cache size:                                  12000
  Current entries:                                13

  Flows added:                                    26
  Flows aged:                                     13
    - Active timeout      (  1800 secs)          1
    - Inactive timeout    (    15 secs)         12

CONN IPV4 INITIATOR ADDR,CONN IPV4 RESPONDER ADDR,CONN RESPONDER
PORT,FLOW OBSPOINT ID,IP VERSION,IP
PROT,APP NAME,flow dirn,time abs first,time abs last,conn initiator,conn
 count new,conn server packets
cnt,conn client packets cnt,conn server network bytes cnt,conn client
network bytes cnt
64.103.125.147,144.254.71.184,53,4294967305,4,17,port
dns,Input,08:55:46.917,08:55:46.917,Initiator,2,1,1,190,106
64.103.121.103,10.1.1.2,67,4294967305,4,17,layer7
dhcp,Input,08:55:47.917,08:55:47.917,Initiator,1,0,1,0,350
64.103.125.3,64.103.125.97,68,4294967305,4,17,layer7
dhcp,Input,08:55:47.917,08:55:53.917,Initiator,1,0,4,0,1412
10.0.2.6,157.55.40.149,443,4294967305,4,6,layer7 ms-
lync,Input,08:55:46.917,08:55:46.917,Initiator,2,10,14,6490,1639
64.103.126.28,66.163.36.139,443,4294967305,4,6,layer7 cisco-jabber-
im,Input,08:55:46.917,08:55:46.917,Initiator,2,12,10,5871,2088
64.103.125.2,64.103.125.29,68,4294967305,4,17,layer7
dhcp,Input,08:55:47.917,08:55:47.917,Initiator,1,0,2,0,712
64.103.125.97,64.103.101.181,67,4294967305,4,17,layer7
dhcp,Input,08:55:47.917,08:55:47.917,Initiator,1,0,1,0,350
192.168.100.6,10.10.20.1,5060,4294967305,4,17,layer7 cisco-jabber-
control,Input,08:55:46.917,08:55:46.917,Initiator,1,0,2,0,2046
64.103.125.3,64.103.125.29,68,4294967305,4,17,layer7
dhcp,Input,08:55:47.917,08:55:47.917,Initiator,1,0,2,0,712
10.80.101.18,10.80.101.6,5060,4294967305,4,6,layer7 cisco-collab-
control,Input,08:55:46.917,08:55:47.917,Initiator,2,23,27,12752,8773
10.1.11.4,66.102.11.99,80,4294967305,4,6,layer7 google-
services,Input,08:55:46.917,08:55:46.917,Initiator,2,3,5,1733,663
```

```
64.103.125.2,64.103.125.97,68,4294967305,4,17,layer7
dhcp,Input,08:55:47.917,08:55:53.917,Initiator,1,0,4,0,1412
64.103.125.29,64.103.101.181,67,4294967305,4,17,layer7
dhcp,Input,08:55:47.917,08:55:47.917,Initiator,1,0,1,0,350
```

基本的なトラブルシューティング：質問と回答

以下に、有線 Application Visibility and Control のトラブルシューティングに関する基本的な質問と回答を示します。

1. **質問**：IPv6 トラフィックが分類されていません。

   回答：現在は IPv4 トラフィックのみがサポートされています。

2. **質問**：マルチキャスト トラフィックが分類されていません。

   回答：現在はユニキャスト トラフィックのみがサポートされています。

3. **質問**：ping を送信したときに、分類されているかを確認できません。

   回答：TCP/UDP プロトコルのみがサポートされています。

4. **質問**：SVI に NBAR を接続できないのはなぜですか。

   回答：NBAR は物理インターフェイスでのみサポートされています。

5. **質問**：ほどんとのトラフィックが CAPWAP トラフィックになっているのですが、なぜですか。

   回答：ワイヤレス アクセス ポートに接続されていないアクセス ポートで NBAR が有効になっていることを確認してください。AP から着信するすべてのトラフィックは capwap として分類されます。この場合、実際の分類は AP または WLC で行われます。

6. **質問**：プロトコル検出で、トラフィックが片側でしか確認できません。さらに、多くの未知のトラフィックがあります。

   回答：これは通常、NBAR が非対称トラフィックを確認していることを示します。片側のトラフィックは 1 つのスイッチ メンバーに分類され、もう一方は別のメンバーに分類されます。トラフィックの両側が確認されるアクセス ポートにのみ NBAR を接続することを推奨します。複数のアップリンクがある場合は、この問題のためそれらに NBAR を接続することはできません。ポート チャネルの一部であるインターフェイスに NBAR を設定した場合にも同様の問題が発生します。

7. **質問**：プロトコル検出で、すべてのアプリケーションの集約ビューが表示されます。時間経過に伴うトラフィック分布を確認するにはどうしたらいいですか。

   回答：WebUI を使用して、過去 48 時間の経時的なトラフィックを表示できます。

8. **質問**：**match protocol** *protocol-name* コマンドを使用してキューベースのイーグレス ポリシーを設定できません。

回答：NBAR2 ベースの分類子が含まれるポリシーでは、**shape** および **set DSCP** のみがサポートされています。一般的な方法としては、入力で DSCP を設定し、DSCP に基づいて出力でシェーピングを実行します。

9. **質問**：インターフェイスに接続している NBAR2 はありませんが、NBAR2 がいまだにアクティブになっています。

   回答：**match protocol** *protocol-name* を含むクラス マップがあると、NBAR はスタックでグローバルにアクティブになりますが、トラフィックは NBAR 分類の対象にはなりません。これは予期された動作であり、リソースを消費しません。

10. **質問**：デフォルトの QOS キューの下にトラフィックがあります。なぜですか。

    回答：新しい各フローでは、フローを分類してハードウェアに結果をインストールするためにいくつかのパケットが使われます。この間に、分類は「不明」となり、トラフィックはデフォルト キューに入ります。

Application Visibility and Control に関する追加情報

関連資料

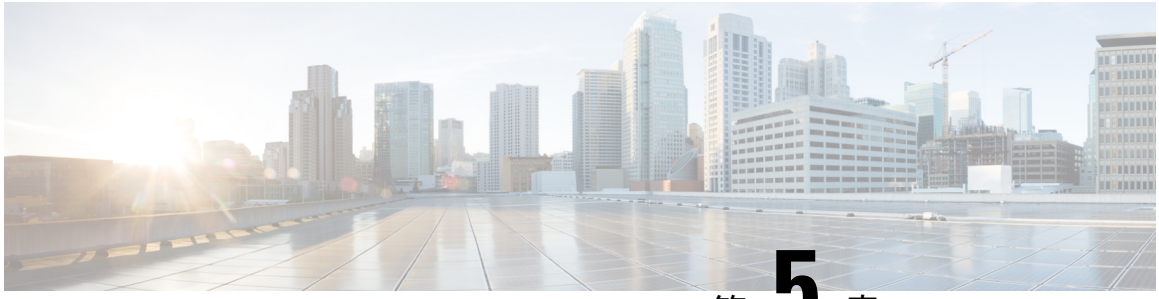
関連項目	参照先
QoS	*NBAR Configuration Guide, Cisco IOS XE Release 16 .x*
NBAR2 プロトコル パック ヒットレス アップグレード	*NBAR Configuration Guide, Cisco IOS XE Release 16.x*

テクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Product Alert Tool（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

有線ネットワークでの Application Visibility and Control の機能履歴と情報

リリース	機能情報
Cisco IOS XE Denali 16.3.2	有線 AVC Flexible NetFlow（FNF）：この機能は、インターフェイスごとのクライアント、サーバ、アプリケーションの統計情報を提供するために、フローレコードでアプリケーション名をキーとして使用します。
Cisco IOS XE Denali 16.3.1	この機能が導入されました。



第 5 章

SDM テンプレートの設定

SDM テンプレートの設定に関する情報

SDM テンプレート

SDM テンプレートを使用してシステム リソースを設定すると、特定の機能に対するサポートをネットワーク内でのデバイスの使用方法に応じて最適化することができます。一部の機能に最大システム使用率を提供するようにテンプレートを選択できます。

テンプレートを変更し、システムを再起動したら、**show sdm prefer** 特権 EXEC コマンドを使用して、新しいテンプレート設定を確認できます。**reload** 特権 EXEC コマンドを入力する前に、**show sdm prefer** コマンドを入力すると、**show sdm prefer** コマンドにより、現在使用しているテンプレートおよびリロード後にアクティブになるテンプレートが表示されます。

デフォルトは Advanced テンプレートです。

（注） SDM テンプレートは VLAN を作成しません。SDM テンプレートにコマンドを追加する前に、VLAN を作成する必要があります。

表には、テンプレートが選択されたときに設定される、おおよそのハードウェア上限が示されています。ハードウェア リソースのある部分がいっぱいの場合は、処理のオーバーフローはすべて CPU に送られ、スイッチのパフォーマンスに重大な影響が出ます。

SDM テンプレートの設定方法

SDM テンプレートの設定

スイッチ SDM テンプレートの設定

SDM テンプレートの設定

SDM テンプレートを使用して機能動作を最適にサポートするには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	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 3	**sdm prefer distribution | nat** 例： `Device(config)# sdm prefer nat`	スイッチで使用する SDM テンプレートを指定します。キーワードの意味は次のとおりです。 • **distribution**：ディストリビューション テンプレートを設定します。 • **nat**：スイッチでの NAT コンフィギュレーションを最大化します。 （注） **no sdm prefer** コマンドとデフォルト テンプレートはサポートされません。
ステップ 4	**end** 例： `Device(config)# end`	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 5	**reload** 例： `Device# reload`	オペレーティング システムをリロードします。 システムの再起動後、**show sdm prefer** 特権 EXEC コマンドを使用して、新しいテンプレート設定を確認できます。reload 特権 EXEC コマンドを入力する前に、**show sdm prefer** コマンドを入力すると、**show sdm prefer** コマンドにより、現在使用しているテンプレートおよびリロード後にアクティブになるテンプレートが表示されます。

SDM テンプレートのモニタリングおよびメンテナンス

コマンド	目的
show sdm prefer	使用中の SDM テンプレートを表示します。
reload	スイッチをリロードして、新しく設定した SDM テンプレートをアクティブにします。
no sdm prefer	デフォルトの SDM テンプレートを設定します。

（注） SDM テンプレートには、テンプレートの一部として定義されているコマンドのみが含まれています。テンプレートで定義されていない別の関連コマンドがテンプレートで有効になっている場合、**show running config** コマンドを入力すると、この他のコマンドが表示されます。たとえば、SDM テンプレートで **switchport voice vlan** コマンドが有効になっている場合、（SDM テンプレートでは定義されていませんが）**spanning-tree portfast edge** コマンドも有効にすることができます。

SDM テンプレートが削除された場合、他の関連するコマンドも削除され、明示的に再設定する必要があります。

SDM テンプレートの設定例

例：SDM テンプレートの表示

次に、詳細なテンプレート情報を表示した出力例を示します。

```
Device# show sdm prefer nat

Showing SDM Template Info

This is the NAT template.
  Number of VLANs:                                4094
  Unicast MAC addresses:                          32768
  Overflow Unicast MAC addresses:                 512
  IGMP and Multicast groups:                      8192
  Overflow IGMP and Multicast groups:             512
  Directly connected routes:                      32768
  Indirect routes:                                65536
  Security Access Control Entries:                18432
  QoS Access Control Entries:                     3072
  Policy Based Routing ACEs:                      16384
  Ingress Netflow ACEs:                           1024
  Egress Netflow ACEs:                            2048
  Flow SPAN ACEs:                                 1024
  Tunnels:                                        1024
  LISP Instance Mapping Entries:                  1024
  Control Plane Entries:                          1024
  Input Netflow flows:                            65536
  Output Netflow flows:                           65536
  SGT/DGT (or) MPLS VPN entries:                  32768
  SGT/DGT (or) MPLS VPN Overflow entries:         512
These numbers are typical for L2 and IPv4 features.
Some features such as IPv6, use up double the entry size;
so only half as many entries can be created.
```

例：SDM テンプレートの設定

```
Device(config)# sdm prefer distribution
Device(config)# exit
Device# reload
  Proceed with reload? [confirm]
```

SDM テンプレートに関する追加情報

関連資料

関連項目	参照先
コマンド リファレンス	*Command Reference (Catalyst 9500 Series Switches)*

標準および RFC

標準/RFC	タイトル
なし	—

MIB

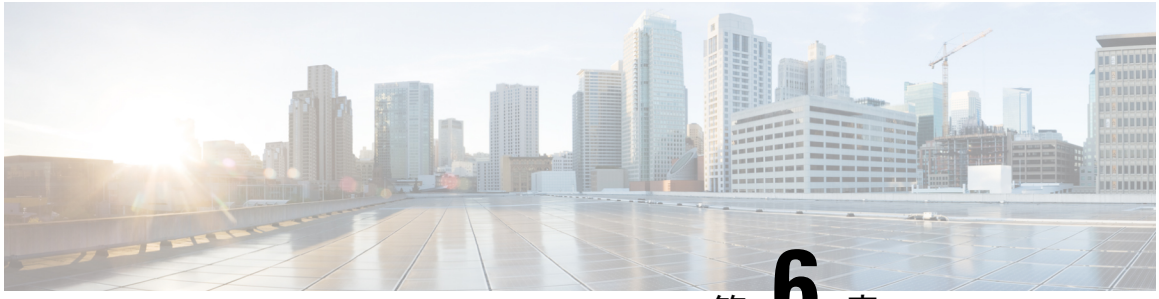
MIB	MIB リンク
本リリースでサポートするすべての MIB	選択したプラットフォーム、Cisco IOS リリース、およびフィーチャ セットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

テクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Product Alert Tool（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

SDM テンプレートの設定の機能履歴と情報

リリース	変更箇所
Cisco IOS XE Everest 16.5.1a	この機能が導入されました。



第 6 章

システム メッセージ ログの設定

システム メッセージ ログの設定に関する情報

システム メッセージ ロギング

スイッチはデフォルトで、システム メッセージおよび **debug** 特権 EXEC コマンドの出力をロギング プロセスに送信します。ロギング プロセスはログ メッセージを各宛先（設定に応じて、ログ バッファ、端末回線、UNIX Syslog サーバなど）に配信する処理を制御します。ロギング プロセスは、コンソールにもメッセージを送信します。

ロギング プロセスがディセーブルの場合、メッセージはコンソールにのみ送信されます。メッセージは生成時に送信されるため、メッセージおよびデバッグ出力にはプロンプトや他のコマンドの出力が割り込みます。メッセージがアクティブなコンソールに表示されるのは、メッセージを生成したプロセスが終了してからです。

メッセージの重大度を設定して、コンソールおよび各宛先に表示されるメッセージのタイプを制御できます。ログ メッセージにタイム スタンプを設定したり、Syslog 送信元アドレスを設定したりして、リアルタイムのデバッグ機能および管理機能を強化できます。表示されるメッセージについては、このリリースに対応するシステム メッセージ ガイドを参照してください。

ロギングされたシステム メッセージにアクセスするには、スイッチのコマンドライン インターフェイス（CLI）を使用するか、または適切に設定された Syslog サーバにこれらのシステム メッセージを保存します。スイッチ ソフトウェアは、Syslog メッセージをスタンドアロン スイッチ上の内部バッファに保存します。スタンドアロン スイッチ、ログをフラッシュ メモリに保存していなかった場合、ログは失われます。

システム メッセージをリモートで監視するには、Syslogサーバ上でログを表示するか、あるいはTelnet、コンソール ポート、またはイーサネット管理ポート経由でスイッチにアクセスします。

（注） Syslog フォーマットは 4.3 Berkeley Standard Distribution（BSD）UNIX と互換性があります。

システム ログ メッセージのフォーマット

システム ログ メッセージは最大 80 文字とパーセント記号（%）、およびその前に配置されるオプションのシーケンス番号やタイム スタンプ情報（設定されている場合）で構成されています。スイッチに応じて、メッセージは次のいずれかの形式で表示されます。

- *seq no:timestamp: %facility-severity-MNEMONIC:description (hostname-n)*
- *seq no:timestamp: %facility-severity-MNEMONIC:description*

パーセント記号の前にあるメッセージの部分は、次のグローバル コンフィギュレーション コマンドの設定によって異なります。

- **service sequence-numbers**
- **service timestamps log datetime**
- **service timestamps log datetime** [**localtime**] [**msec**] [**show-timezone**]
- **service timestamps log uptime**

表 *10*: システム ログ メッセージの要素

要素	説明
seq no:	**service sequence-numbers** グローバル コンフィギュレーション コマンドが設定されている場合だけ、ログ メッセージにシーケンス番号をスタンプします。
timestamp のフォーマット： *mm/dd hh:mm:ss* または *hh:mm:ss*（短時間） または *d h*（長時間）	メッセージまたはイベントの日時です。**service timestamps log** [**datetime** | **log**] グローバル コンフィギュレーション コマンドが設定されている場合だけ、この情報が表示されます。
facility	メッセージが参照する機能（SNMP、SYS など）です。
severity	メッセージの重大度を示す 0 ～ 7 の 1 桁のコードです。

要素	説明
MNEMONIC	メッセージを一意に示すテキスト ストリングです。
説明	レポートされているイベントの詳細を示すテキスト ストリングです。

デフォルトのシステム メッセージ ロギングの設定

表 *11* : デフォルトのシステム メッセージ ロギングの設定

機能	デフォルト設定
コンソールへのシステム メッセージ ロギング	有効。
コンソールの重大度	デバッグ
ログ ファイル設定	ファイル名の指定なし
ログ バッファ サイズ	4096 バイト
ログ履歴サイズ	1 メッセージ
タイム スタンプ	ディセーブル。
同期ロギング	ディセーブル。
ロギング サーバ	ディセーブル。
Syslog サーバの IP アドレス	未設定
サーバ機能	Local7
サーバの重大度	通知

syslog メッセージの制限

snmp-server enable trap グローバル コンフィギュレーション コマンドを使用して、SNMP ネットワーク管理ステーション（NMS）に送信されるように Syslog メッセージ トラップがイネーブルに設定されている場合は、スイッチの履歴テーブルに送信および格納されるメッセージの重大度を変更できます。また、履歴テーブルに格納されるメッセージの数を変更することもできます。

SNMP トラップは宛先への到達が保証されていないため、メッセージは履歴テーブルに格納されます。デフォルトでは、Syslog トラップが有効でない場合も、レベルが **warning** であるメッセージや数値的に下位レベルのメッセージの 1 つが履歴テーブルに格納されます。

履歴テーブルがいっぱいの場合（**logging history size** グローバル コンフィギュレーション コマンドで指定した最大メッセージ エントリ数が格納されている場合）は、新しいメッセージ エントリを格納できるように、最も古いエントリがテーブルから削除されます。

履歴テーブルは、level キーワードおよび重大度を示します。SNMP を使用している場合は、重大度の値が 1 だけ増えます。たとえば、*emergencies* は 0 ではなく 1 に、*critical* は 2 ではなく 3 になります。

システム メッセージ ログの設定方法

メッセージ表示宛先デバイスの設定

メッセージロギングがイネーブルの場合、コンソールだけでなく特定の場所にもメッセージを送信できます。

このタスクはオプションです。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	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	**logging buffered** [*size*] 例： `Device(config)# logging buffered 8192`	スイッチ上で、ログ メッセージを内部バッファに保存します。有効な範囲は 4096 ～ 2147483647 バイトです。デフォルトのバッファ サイズは 4096 バイトです。 スタンドアロン スイッチに障害が発生すると、ログ ファイルをフラッシュ メモリに保存していなかった場合、ログ ファイルは失われます。ステップ 4 を参照してください。

	コマンドまたはアクション	目的
		（注） バッファ サイズを大きすぎる値に設定しないでください。他の作業に使用するメモリが不足することがあります。スイッチ上の空きプロセッサ メモリを表示するには、**show memory** 特権 EXEC コマンドを使用します。ただし、表示される値は使用できる最大値であるため、バッファ サイズをこの値に設定しないでください。
ステップ 3	**logging** ホスト 例： `Device(config)# logging 125.1.1.100`	UNIX Syslog サーバ ホストにメッセージを保存します。 *host* には、syslog サーバとして使用するホストの名前または IP アドレスを指定します。 ログ メッセージを受信する Syslog サーバのリストを作成するには、このコマンドを複数回入力します。
ステップ 4	**logging file flash:** *filename* [*max-file-size* [*min-file-size*]] [*severity-level-number* | *type*] 例： `Device(config)# logging file flash:log_msg.txt 40960 4096 3`	スタンドアロン スイッチ上で、フラッシュ メモリにあるファイルにログ メッセージを保存します。 • *filename*：ログ メッセージのファイル名を入力します。 • （任意）**max-file-size**：ログ ファイルの最大サイズを指定します。指定できる範囲は 4096 ～ 2147483647 です。デフォルトは 4096 バイトです。 • （任意）*min-file-size*：ログ ファイルの最小サイズを指定します。指定できる範囲は 1024 ～ 2147483647 です。デフォルトは 2048 バイトです。 • （任意）*severity-level-number* | *type*：ロギングの重大度またはロギング タイプを指定します。重大度に指定できる範囲は 0 ～ 7 です。

	コマンドまたはアクション	目的
ステップ 5	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 6	**terminalmonitor** 例： `Device# terminal monitor`	現在のセッション間、非コンソール端末にメッセージを保存します。 端末パラメータ コンフィギュレーション コマンドはローカルに設定され、セッションの終了後は無効になります。デバッグ メッセージを表示する場合は、セッションごとにこのステップを実行する必要があります。

ログ メッセージの同期化

特定のコンソール ポート回線または仮想端末回線に対して、非送信請求メッセージおよび **debug** 特権 EXEC コマンドの出力を送信請求デバイスの出力およびプロンプトと同期させることができます。重大度に応じて非同期に出力されるメッセージのタイプを特定できます。また、端末の非同期メッセージが削除されるまで保存しておくバッファの最大数を設定することもできます。

非送信請求メッセージおよび **debug** コマンド出力の同期ロギングがイネーブルの場合、送信請求デバイス出力がコンソールに表示または印刷された後に、非送信請求デバイスからの出力が表示または印刷されます。非送信請求メッセージおよび **debug** コマンドの出力は、ユーザ入力用プロンプトが返された後に、コンソールに表示されます。したがって、非送信請求メッセージおよび **debug** コマンドの出力は、送信請求デバイス出力およびプロンプトに割り込まれることはありません。非送信請求メッセージが表示された後に、コンソールはユーザ プロンプトを再表示します。

このタスクはオプションです。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	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	**line** [**console** | **vty**] *line-number* [*ending-line-number*] 例： `Device(config)# line console`	メッセージの同期ロギングに設定する回線を指定します。 • **console**：スイッチ コンソール ポートまたはイーサネット管理ポートでの設定を指定します。 • **line vty** *line-number*：どの vty 回線の同期ロギングをイネーブルにするかを指定します。Telnet セッションを介して行われる設定には、vty 接続を使用します。回線番号に指定できる範囲は 0 ～ 15 です。 16 個の vty 回線の設定をすべて一度に変更するには、次のように入力します。 line vty 0 15 また、現在の接続に使用されている 1 つの vty 回線の設定を変更することもできます。たとえば、vty 回線 2 の設定を変更するには、次のように入力します。 line vty 2 このコマンドを入力すると、ライン コンフィギュレーション モードになります。
ステップ 3	**logging synchronous** [**level** [*severity-level* | **all**] | **limit** *number-of-buffers*] 例： `Device(config)# logging synchronous level 3 limit 1000`	メッセージの同期ロギングをイネーブルにします。 • （任意）**level** *severity-level*：メッセージの重大度レベルを指定します。重大度がこの値以上であるメッセージは、非同期に出力されます。値が小さいほど重大度は大きく、値が大きいほど重大度は小さくなります。デフォルトは 2 です。 • （任意）**level all**：重大度に関係なく、すべてのメッセージが非同期に出力されます。 • （任意）**limit** *number-of-buffers*：キューイングされる端末のバッファ数を指定します。これを超える新しいメッセージは廃棄されます。指定

	コマンドまたはアクション	目的
		できる範囲は 0 ～ 2147483647 です。デフォルトは 20 です。
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

メッセージ ロギングのディセーブル化

メッセージロギングはデフォルトでイネーブルに設定されています。コンソール以外のいずれかの宛先にメッセージを送信する場合は、メッセージロギングをイネーブルにする必要があります。メッセージ ロギングがイネーブルの場合、ログ メッセージはロギング プロセスに送信されます。ロギングプロセスは、メッセージを生成元プロセスと同期しないで指定場所に記録します。

ロギングプロセスをディセーブルにすると、メッセージがコンソールに書き込まれるまでプロセスは処理続行を待機する必要があるため、スイッチの処理速度が低下することがあります。ロギングプロセスがディセーブルの場合、メッセージは生成後すぐに（通常はコマンド出力に割り込む形で）コンソールに表示されます。

logging synchronous グローバル コンフィギュレーション コマンドも、コンソールへのメッセージ表示に影響します。このコマンドをイネーブルにすると、Return を押さなければメッセージが表示されません。

メッセージ ロギングをディセーブルにした後に再びイネーブルにするには、**logging on** グローバル コンフィギュレーション コマンドを使用します。

このタスクはオプションです。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	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	**no logging console** 例： `Device(config)# no logging console`	メッセージ ロギングをディセーブルにします。

	コマンドまたはアクション	目的
ステップ 3	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

ログ メッセージのタイム スタンプのイネーブル化およびディセーブル化

デフォルトでは、ログ メッセージにはタイム スタンプが適用されません。

このタスクはオプションです。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	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	次のいずれかのコマンドを使用します。
	• **servicetimestampsloguptime**
	• **service timestamps log datetime**[**msec** | **localtime** | **show-timezone**]
	例：
	`Device(config)# service timestamps log uptime`
	または
	`Device(config)# service timestamps log datetime`
		ログのタイムスタンプをイネーブルにします。
		• **log uptime**：ログ メッセージのタイムスタンプをイネーブルにして、システムの再起動以降の経過時間を表示します。
		• **log datetime**：ログ メッセージのタイムスタンプをイネーブルにします。選択したオプションに応じて、ローカル タイム ゾーンを基準とした日付、時間（ミリ秒）、タイム ゾーン名をタイム スタンプとして表示できます。

ステップ 3	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

ログメッセージのシーケンス番号のイネーブル化およびディセーブル化

タイムスタンプが同じログ メッセージが複数ある場合、これらのメッセージを表示するには、シーケンス番号を使用してメッセージを表示できます。デフォルトでは、ログ メッセージにシーケンス番号は表示されません。

このタスクはオプションです。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	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	**service sequence-numbers** 例： `Device(config)# service sequence-numbers`	シーケンス番号をイネーブルにします。
ステップ 3	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

メッセージ重大度の定義

メッセージの重大度を指定して、選択したデバイスに表示されるメッセージを制限します。

このタスクはオプションです。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	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	**logging console** *level* 例： `Device(config)# logging console 3`	コンソールに保存するメッセージを制限します。 デフォルトで、コンソールはデバッグメッセージ、および数値的により低いレベルのメッセージを受信します。
ステップ 3	**logging monitor** *level* 例： `Device(config)# logging monitor 3`	端末回線に出力するメッセージを制限します。 デフォルトで、端末はデバッグメッセージ、および数値的により低いレベルのメッセージを受信します。
ステップ 4	**logging trap** *level* 例： `Device(config)# logging trap 3`	Syslog サーバに保存するメッセージを制限します。 デフォルトで、Syslog サーバは通知メッセージ、および数値的により低いレベルのメッセージを受信します。
ステップ 5	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

履歴テーブルおよび SNMP に送信される syslog メッセージの制限

このタスクでは、履歴テーブルおよび SNMP に送信される syslog メッセージを制限する方法について説明します。

このタスクはオプションです。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	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	**logging history** *level* 例： `Device(config)# logging history 3`	履歴ファイルに保存され、SNMP サーバに送信される syslog メッセージのデフォルト レベルを変更します。 デフォルトでは、**warnings**、**errors**、**critical**、**alerts**、および **emergencies** のメッセージが送信されます。
ステップ 3	**logging history size** *number* 例： `Device(config)# logging history size 200`	履歴テーブルに保存できる Syslog メッセージの数を指定します。 デフォルトでは1つのメッセージが格納されます。指定できる範囲は0～500です。
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

UNIX Syslog デーモンへのメッセージのロギング

このタスクはオプションです。

（注） 最新バージョンの UNIX Syslog デーモンの中には、デフォルトでネットワークからの Syslog パケットを受け入れないものがあります。このようなシステムの場合に、Syslog メッセージのリモート ロギングをイネーブルにするには、Syslog コマンド ラインに追加または削除する必要があるオプションを、UNIX の **man syslogd** コマンドを使用して判別します。

始める前に

- root としてログインします。
- システム ログ メッセージを UNIX Syslog サーバに送信する前に、UNIX サーバ上で Syslog デーモンを設定する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	/etc/syslog.conf ファイルに次の行を追加します。	• **local7**：ロギング機能を指定します。

	コマンドまたはアクション	目的
	例： **local7.debug /usr/adm/logs/***cisco.log*	• **debug**：syslog レベルを指定します。このファイルは、syslogデーモンに書き込み権限がある既存ファイルである必要があります。
ステップ 2	UNIX シェルプロンプトに次のコマンドを入力します。 例： $ **touch /var/log/***cisco.log* $ **chmod 666 /var/log/***cisco.log*	ログファイルを作成します。syslogデーモンは、このレベルまたはこのファイルのより高い重大度レベルでメッセージを送信します。
ステップ 3	Syslogデーモンに新しい設定を認識させます。 例： $ **kill -HUP `cat /etc/syslog.pid`**	詳細については、ご使用のUNIXシステムの **man syslog.conf** および **man syslogd** コマンドを参照してください。

システムメッセージログのモニタリングおよびメンテナンス

コンフィギュレーション アーカイブ ログのモニタリング

コマンド	目的
show archive log config {**all** | **number** [*end-number*] | **user** *username* [**session** *number*] *number* [*end-number*] | **statistics**} [**provisioning**]	コンフィギュレーションログ全体、または指定されたパラメータのログを表示します。

システム メッセージ ログの設定例

例：スイッチ システム メッセージ

次に、スイッチ上のスイッチ システム メッセージの一部を示します。

```
00:00:46: %LINK-3-UPDOWN: Interface Port-channel1, changed state to up
00:00:47: %LINK-3-UPDOWN: Interface GigabitEthernet0/1, changed state to up
00:00:47: %LINK-3-UPDOWN: Interface GigabitEthernet0/2, changed state to up
00:00:48: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to down
00:00:48: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed
state to down 2
*Mar  1 18:46:11: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
18:47:02: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
*Mar  1 18:48:50.483 UTC: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
```

システム メッセージ ログに関する追加情報

関連資料

関連項目	参照先
システム管理コマンド	*Command Reference (Catalyst 9500 Series Switches)*

標準および RFC

標準/RFC	タイトル
なし	—

MIB

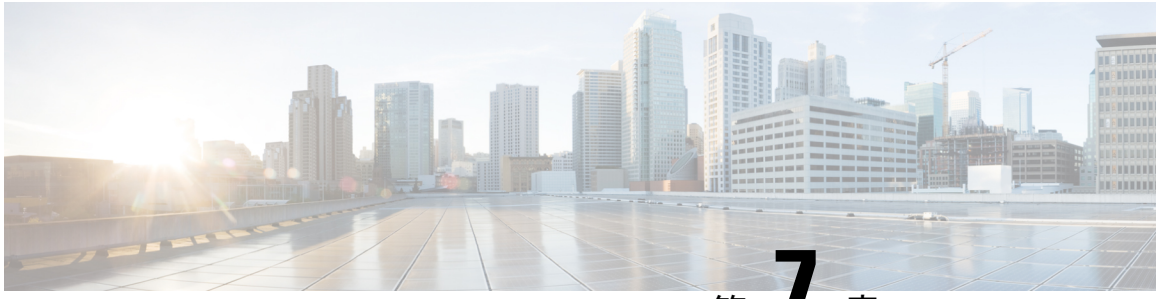
MIB	MIB リンク
本リリースでサポートするすべての MIB	選択したプラットフォーム、Cisco IOS リリース、およびフィーチャ セットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

テクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Product Alert Tool（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

システム メッセージ ログの機能履歴と情報

リリース	変更箇所
Cisco IOS XE Everest 16.5.1a	この機能が導入されました。



第 7 章

オンライン診断の設定

オンライン診断の設定に関する情報

オンライン診断

オンライン診断では、デバイスが稼働中のネットワークに接続している間に、デバイスのハードウェア機能をテストし、確認できます。

オンライン診断には、異なるハードウェア コンポーネントをチェックするパケット交換テストが含まれ、データ パスおよび制御信号が確認されます。

オンライン診断では、次の領域の問題が検出されます。

- ハードウェア コンポーネント
- インターフェイス（イーサネット ポートなど）
- はんだ接合

オンライン診断は、オンデマンド診断、スケジュール診断、ヘルスモニタリング診断に分類できます。オンデマンド診断は、CLI から実行されます。スケジュールされた診断は、動作中のネットワークにデバイスが接続されているときに、ユーザが指定した間隔または指定した時刻に実行されます。ヘルスモニタリングは、バックグラウンドでユーザが指定した間隔で実行されます。デフォルトでは、30 秒ごとにヘルス モニタリング テストが実行されます。

オンライン診断を設定したあと、手動で診断テストを開始したり、テスト結果を表示したりできます。また、デバイスに設定されているテストの種類、およびすでに実行された診断テスト名を確認できます。

オンライン診断の設定方法

オンライン診断テストの開始

スイッチで実行する診断テストを設定しデバイス、**diagnostic start** 特権 EXEC コマンドを使用して診断テストを開始します。

テストを開始したら、テスト プロセスの停止はできません。

手動でオンライン診断テストを開始するには、次の特権 EXEC コマンドを使用します。

手順

	コマンドまたはアクション	目的
ステップ 1	**diagnostic start switch** *number* **test** {*name* | *test-id* | *test-id-range* | **all** | **basic** | **complete** | **minimal** | **non-disruptive** | **per-port**} 例： `Device# diagnostic start switch 2 test basic`	診断テストを開始します。 次のいずれかのオプションを使用してテストを指定できます。 • *name*：テストの名前を入力します。 • *test-id*：テストの ID 番号を入力します。 • *test-id-range*：カンマとハイフンで区切ってテスト ID の範囲を整数で入力します。 • **all**：すべてのテストを開始します。 • **basic**：基本テスト スイートを開始します。 • **complete**：完全なテスト スイートを開始します。 • **minimal**：最小限のブートアップ テスト スイートを開始します。 • **non-disruptive**：ノンディスラプティブ テスト スイートを開始します。 • **per-port**：ポート単位のテスト スイートを開始します。

オンライン診断の設定

診断モニタリングをイネーブルにする前に、障害しきい値およびテストの間隔を設定する必要があります。

オンライン診断のスケジューリング

特定のデバイスについて指定した時間、または日、週、月単位でオンライン診断をスケジューリングできます。スケジューリングを削除するには、コマンドの **no** 形式を入力します。

手順

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 例： `Device# configure terminal`	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	**diagnostic schedule switch** *number***test** {*name* | *test-id* | *test-id-range* | **all** | **basic** | **complete** | **minimal** | **non-disruptive** | **per-port**} {**daily** | **on** *mm dd yyyy hh:mm* | **port** *inter-port-number port-number-list* | **weekly** *day-of-week hh:mm*} 例： `Device(config)# diagnostic schedule switch 3 test 1-5 on July 3 2013 23:10`	特定日時のオンデマンド診断テストをスケジュールします。 スケジュールするテストを指定する場合は、次のオプションを使用します。 • *name*：**show diagnostic content** コマンドの出力に表示されるテストの名前です。 • *test-id*：**show diagnostic content** コマンドの出力に表示されるテストの ID 番号です。 • *test-id-range*：**show diagnostic content** コマンドの出力に表示されるテストの ID 番号です。 • **all**：すべてのテスト ID • **basic**：基本的なオンデマンドの診断テストを開始します。 • **complete**：完全なテスト スイートを開始します。 • **minimal**：最小限のブートアップテスト スイートを開始します。

	コマンドまたはアクション	目的
		• **non-disruptive**：ノンディスラプティブ テスト スイートを開始します。 • **per-port**：ポート単位のテスト スイートを開始します。 テストは次のようにスケジュールできます。 • 毎日：**daily** *hh:mm* パラメータを使用します。 • 特定日時：**on** *mm dd yyyy hh:mm* パラメータを使用します。 • 毎週：**weekly** *day-of-week hh:mm* パラメータを使用します。

ヘルス モニタリング診断の設定

デバイスが稼働中のネットワークに接続されている間に、スイッチに対しヘルスモニタリング診断テストを設定できます。ヘルス モニタリング テストの実行間隔を設定したり、テスト失敗時のデバイスのsyslogメッセージ生成をイネーブルにしたり、特定のテストをイネーブルにできます。

テストをディセーブルにするには、コマンドの **no** 形式を入力します。

デフォルトでは、ヘルス モニタリングはディセーブルですが、デバイスはテストの失敗時にSyslog メッセージを生成します。

ヘルス モニタリング診断テストを設定し、イネーブルにするには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	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 3	**diagnostic monitor intervalswitch***number***test** {*name* | *test-id* | *test-id-range* | **all**} *hh:mm:ss milliseconds day* 例： `Device(config)# diagnostic monitor interval switch 2 test 1 12:30:00 750 5`	指定のテストに対し、ヘルスモニタリングの実行間隔を設定します。 テストを指定する場合は、次のいずれかのパラメータを使用します。 • *name*：**show diagnostic content** コマンドの出力に表示されるテストの名前です。 • *test-id*：**show diagnostic content** コマンドの出力に表示されるテストの ID 番号です。 • *test-id-range*：**show diagnostic content** コマンドの出力に表示されるテストの ID 番号です。 • **all**：すべての診断テスト。 間隔を指定する場合は、次のパラメータを設定します。 • *hh:mm:ss*：モニタリング間隔（時間、分、秒）。指定できる範囲は *hh* が 0 ~ 24、*mm* および *ss* が 0 ~ 60 です。 • *milliseconds*：モニタリング間隔（ミリ秒（ms））。範囲は 0 ～ 999 です。 • *day*：モニタリング間隔（日数）。範囲は 0 ～ 20 です。
ステップ 4	**diagnostic monitor syslog** 例： `Device(config)# diagnostic monitor syslog`	（任意）ヘルスモニタリングテストの失敗時にスイッチが Syslog メッセージを生成するように設定します。
ステップ 5	**diagnostic monitor thresholdswitch** *number number***test** {*name* | *test-id* | *test-id-range* | **all**} **failure count** *count* 例： `Device(config)# diagnostic monitor`	（任意）ヘルスモニタリングテストの失敗しきい値を設定します。 テストを指定する場合は、次のいずれかのパラメータを使用します。

	コマンドまたはアクション	目的
	`threshold switch 2 test 1 failure count 20`	• *name*：**show diagnostic content** コマンドの出力に表示されるテストの名前です。 • *test-id*：**show diagnostic content** コマンドの出力に表示されるテストの ID 番号です。 • *test-id-range*：**show diagnostic content** コマンドの出力に表示されるテストの ID 番号です。 • **all**：すべての診断テスト。 失敗しきい値 *count* に指定できる範囲は 0 ～ 99 です。
ステップ 6	**diagnostic monitorswitch***number***test** {*name* | *test-id* | *test-id-range* | **all**} 例： `Device(config)# diagnostic monitor switch 2 test 1`	指定のヘルス モニタリングテストをイネーブルにします。 **switch** *number* キーワードは、スタック構成スイッチだけでサポートされます。 テストを指定する場合は、次のいずれかのパラメータを使用します。 • *name*：**show diagnostic content** コマンドの出力に表示されるテストの名前です。 • *test-id*：**show diagnostic content** コマンドの出力に表示されるテストの ID 番号です。 • *test-id-range*：**show diagnostic content** コマンドの出力に表示されるテストの ID 番号です。 • **all**：すべての診断テスト。
ステップ 7	**end** 例： `Device(config)# end`	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 8	**show diagnostic** { **content** | **post** | **result** | **schedule** | **status** | **switch** }	オンライン診断のテスト結果およびサポートされるテストスイートを表示します。
ステップ 9	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 10	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（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。

オンライン診断のモニタリングおよびメンテナンス

オンライン診断テストとテスト結果の表示

デバイスまたはデバイス スタックに設定されているオンライン診断テストを表示し、この表に示す **show** 特権 EXEC コマンドを使用してテスト結果を確認することができます。

表 *12 :* 診断テストの設定および結果用のコマンド

コマンド	目的
show diagnostic contentswitch [*number* | **all**] **show diagnostic content**	スイッチに対して設定されたオンライン診断を表示します。
show diagnostic status	現在実行中の診断テストを表示します。
show diagnostic resultswitch [*number* | **all**] [**detail** | **test** {*name* | *test-id* | *test-id-range* | **all**} [**detail**]]	オンライン診断テストの結果を表示します。
show diagnostic switch [*number* | **all**] [**detail**] **show diagnosticdetail**]	オンライン診断テストの結果を表示します。
show diagnostic schedule [*number* | **all**]	オンライン診断テストのスケジュールを表示します。

コマンド	目的
show diagnostic post	POST結果を表示します（この出力は、**show post** コマンドの出力と同じです）。

オンライン診断テストの設定例

例：診断テストの開始

次に、テスト名を指定して診断テストを開始する例を示します。

```
Device# diagnostic start switch 2 test DiagFanTest
```

次に、すべての基本診断テストを開始する例を示します。

```
Device# diagnostic start switch 1 test all
```

例：ヘルス モニタリング テストの設定

次に、ヘルス モニタリング テストを設定する例を示します。

```
Device(config)# diagnostic monitor threshold switch 1 test 1 failure count 50
Device(config)# diagnostic monitor interval switch 1 test TestPortAsicStackPortLoopback
```

例：診断テストのスケジューリング

次に、特定のスイッチに対して、特定の日時に診断テストを実行するようにスケジューリングする例を示します。

```
Device(config)# diagnostic schedule test DiagThermalTest on June 3 2013  22:25
```

次の例では、指定されたスイッチで毎週特定の時間に診断テストを実行するようにスケジューリングする方法を示します。

```
Device(config)# diagnostic schedule switch 1 test 1,2,4-6 weekly saturday 10:30
```

例：オンライン診断の表示

次に、オンデマンド診断設定を表示する例を示します。

```
Device# show diagnostic ondemand settings

Test iterations = 1
Action on test failure = continue
```

次に、障害の診断イベントを表示する例を示します。

```
Device# show diagnostic events event-type error

Diagnostic events (storage for 500 events, 0 events recorded)
Number of events matching above criteria = 0

No diagnostic log entry exists.
```

次に、診断テストの説明を表示する例を示します。

```
Device# show diagnostic description switch 1 test all

DiagGoldPktTest :
        The GOLD packet Loopback test verifies the MAC level loopback
        functionality. In this test, a GOLD packet, for which doppler
        provides the support in hardware, is sent. The packet loops back
        at MAC level and is matched against the stored packet. It is a non
        -disruptive test.

DiagThermalTest :
        This test verifies the temperature reading from the sensor is below the yellow
        temperature threshold. It is a non-disruptive test and can be run as a health
monitoring test.

DiagFanTest :
        This test verifies all fan modules have been inserted and working properly on
the board
        It is a non-disruptive test and can be run as a health monitoring test.

DiagPhyLoopbackTest :
        The PHY Loopback test verifies the PHY level loopback
        functionality. In this test, a packet is sent which loops back
        at PHY level and is matched against the stored packet. It is a
        disruptive test and cannot be run as a health monitoring test.

DiagScratchRegisterTest :
        The Scratch Register test monitors the health of application-specific
        integrated circuits (ASICs) by writing values into registers and reading
        back the values from these registers. It is a non-disruptive test and can
        be run as a health monitoring test.

DiagPoETest :
        This test checks the PoE controller functionality. This is a disruptive test
        and should not be performed during normal switch operation.


DiagMemoryTest :
        This test runs the exhaustive ASIC memory test during normal switch operation
        NG3K utilizes mbist for this test. Memory test is very disruptive
        in nature and requires switch reboot after the test.

Device#
```

次に、ブートアップ レベルを表示する例を示します。

```
Device# show diagnostic bootup level

Current bootup diagnostic level: minimal

Device#
```

オンライン診断に関する追加情報

関連資料

関連項目	参照先
システム管理コマンド	*Command Reference (Catalyst 9500 Series Switches)*

MIB

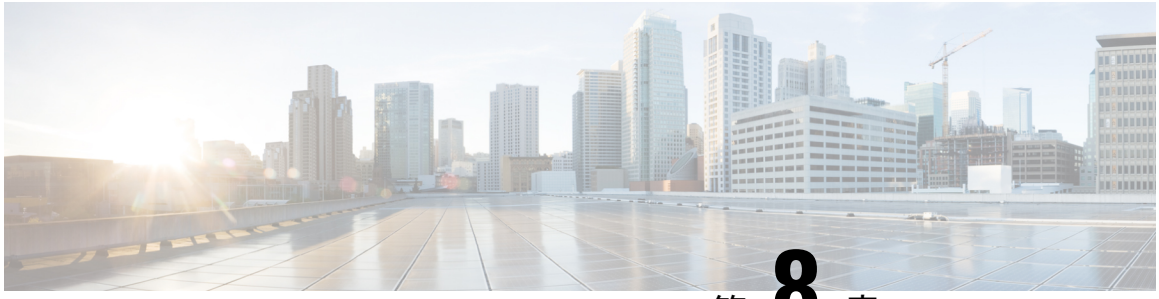
MIB	MIB リンク
本リリースでサポートするすべての MIB	選択したプラットフォーム、Cisco IOS リリース、およびフィーチャ セットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

テクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Product Alert Tool（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

オンライン診断設定の機能履歴と情報

リリース	変更箇所
Cisco IOS XE Everest 16.5.1a	この機能が導入されました。



第 8 章

コンフィギュレーション ファイルの管理

コンフィギュレーション ファイルの管理の前提条件

- ユーザには、少なくとも Cisco IOS 環境とコマンドライン インターフェイスに関する基本的な知識が必要です。
- システムでは、少なくとも最小限の設定が実行されていることが必要です。基本コンフィギュレーション ファイルは、**setup** コマンドを使用して作成できます。

コンフィギュレーション ファイルの管理の制約事項

- このドキュメントで説明されている Cisco IOS コマンドの多くが使用可能であり機能するのは、デバイスの特定のコンフィギュレーション モードでのみです。
- Cisco IOS コンフィギュレーション コマンドのいくつかは、特定のデバイス プラットフォームでのみ使用可能であり、コマンド構文はプラットフォームによって異なる可能性があります。

コンフィギュレーション ファイルの管理について

コンフィギュレーション ファイルのタイプ

コンフィギュレーションファイルには、Ciscoデバイスの機能をカスタマイズするためのCisco IOS ソフトウェア コマンドが含まれています。コマンドは、システムを起動したとき（startup-configファイルから）、またはコンフィギュレーションモードでCLIにコマンドを入力したときに、Cisco IOS ソフトウェアによって解析（変換および実行）されます。

スタートアップ コンフィギュレーションファイル（startup-config）は、ソフトウェアを設定するためにシステムの起動時に使用されます。実行コンフィギュレーション ファイル（running-config）には、ソフトウェアの現在の設定が含まれています。2つのコンフィギュレーションファイルは別々の設定にできます。たとえば、コンフィギュレーションを永続的ではなく短期間で変更する場合があります。この場合、**configureterminal** EXEC コマンドを使用して実行コンフィギュレーションを変更しますが、そのコンフィギュレーションは **copy running-config startup-config** EXEC コマンドを使用して保存しません。

実行コンフィギュレーションを変更するには、コンフィギュレーションファイルの変更（173ページ）の項で説明されているように、**configure terminal** コマンドを使用します。Cisco IOS コンフィギュレーションモードの使用時には、通常コマンドはすぐに実行され、入力直後またはコンフィギュレーション モードを終了した時点で実行コンフィギュレーション ファイルに保存されます。

スタートアップ コンフィギュレーション ファイルを変更するには、**copy running-config startup-config** EXEC コマンドを使用してスタートアップ コンフィギュレーションに実行コンフィギュレーションファイルを保存するか、ファイルサーバからスタートアップコンフィギュレーションにコンフィギュレーション ファイルをコピーします（詳細については、TFTP サーバからデバイスへのコンフィギュレーション ファイルのコピーを参照してください）。

コンフィギュレーション モードおよびコンフィギュレーション ソースの選択

デバイス上でコンフィギュレーションモードを開始するには、特権EXECプロンプトで**configure** コマンドを入力します。Cisco IOS ソフトウェアは次のプロンプトで応答し、端末、メモリ、またはネットワーク サーバ（ネットワーク）上に格納されたファイルのいずれかを、コンフィギュレーション コマンドのソースとして指定するように要求されます。

```
Configuring from terminal, memory, or network [terminal]?
```

端末からの設定では、コマンドラインにコンフィギュレーションコマンドを入力できます（次の項を参照してください）。詳細については、スタートアップ コンフィギュレーション ファイルでのコンフィギュレーション コマンドの再実行の項を参照してください。

ネットワークからの設定では、ネットワーク経由でコンフィギュレーションコマンドをロードして実行できます。詳細については、TFTP サーバからデバイスへのコンフィギュレーション ファイルのコピーの項を参照してください。

CLI を使用したコンフィギュレーション ファイルの変更

Cisco IOS ソフトウェアは、1 行につき 1 つのコンフィギュレーション コマンドを受け入れます。コンフィギュレーションコマンドは、必要なだけ入力できます。コンフィギュレーション ファイルには、入力したコマンドを説明するコメントを追加できます。コメントの先頭には、感嘆符（!）を付けます。コメントは NVRAM にもコンフィギュレーション ファイルのアクティブ コピーにも格納されないため、**show running-config** または **more system:running-config** EXEC コマンドでアクティブな設定のリストを表示しても、コメントは表示されません。**show startup-config** または **more nvram:startup-config** EXEC モード コマンドでスタートアップ コンフィギュレーションのリストを表示しても、コメントは表示されません。コメントは、コンフィギュレーション ファイルがデバイスにロードされたときにコンフィギュレーション ファイルから削除されます。ただし、ファイル転送プロトコル（FTP）、リモート コピープロトコル（RCP）、または Trivial File Transfer Protocol（TFTP）サーバ上に格納されているコンフィギュレーション ファイルのコメントのリストは表示できます。CLI を使用してソフトウェアは設定するときは、ユーザの入力に従ってソフトウェアによりコマンドが実行されます。

コンフィギュレーション ファイルの場所

コンフィギュレーション ファイルは、次の場所に格納されます。

- 実行コンフィギュレーションは RAM に格納されます。
- クラス A フラッシュ ファイル システム プラットフォーム以外のすべてのプラットフォーム上では、スタートアップ コンフィギュレーションは不揮発性 RAM（NVRAM）に格納されます。
- クラス A フラッシュ ファイル システムのプラットフォーム上では、スタートアップ コンフィギュレーションは CONFIG_FILE 環境変数で指定された場所に格納されます（クラス A フラッシュ ファイル システムでの CONFIG_FILE 環境変数の指定 （197 ページ）の項を参照してください）。CONFIG_FILE 変数は、デフォルトでは NVRAM になりますが、次のファイル システムのファイルも指定できます。
  - **nvram:** （NVRAM）
  - **flash:** （内部フラッシュ メモリ）
  - **usbflash0:** （外部 usbflash ファイル システム）

ネットワーク サーバからデバイスへのコンフィギュレーション ファイルのコピー

TFTP、rcp、または FTP サーバからデバイスの実行コンフィギュレーションまたはスタートアップ コンフィギュレーションへコンフィギュレーション ファイルをコピーできます。この機能は、次のいずれかの理由により実行する場合があります。

- バックアップ コンフィギュレーション ファイルを復元するため。
- 別のデバイスにコンフィギュレーションファイルを使用するため。たとえば、別のデバイスをネットワークに追加して、そのコンフィギュレーションを元のデバイスと同様にする場合です。新しいデバイスにファイルをコピーすることにより、ファイル全体を再作成するのではなく、該当部分を変更できます。
- 同一のコンフィギュレーション コマンドをネットワーク内のすべてのデバイスにロードして、すべてのデバイスのコンフィギュレーションを同様にするため。

コマンドラインにコマンドを入力した場合と同様に、**copy**{**ftp:**|**rcp:**|**tftp:system:running-config**} EXEC コマンドはデバイスにコンフィギュレーションファイルをロードします。コマンドを追加する前に、デバイスにより既存の実行コンフィギュレーションが消去されることはありません。コピーされたコンフィギュレーションファイル内のコマンドによって既存のコンフィギュレーションファイル内のコマンドが置き換えられると、既存のコマンドは消去されます。たとえば、コピーされたコンフィギュレーション ファイルに格納されている特定のコマンドの IP アドレスが、既存のコンフィギュレーションに格納されている IP アドレスと異なる場合は、コピーされたコンフィギュレーション内の IP アドレスが使用されます。ただし、既存のコンフィギュレーション内の一部のコマンドには、置き換えられたり無効になったりしないものもあります。このようなコマンドがある場合は、既存のコンフィギュレーション ファイルとコピーされたコンフィギュレーション ファイルが組み合わされた（コピーされたコンフィギュレーション ファイルが優先する）コンフィギュレーション ファイルが作成されます。

コンフィギュレーションファイルをサーバ上に格納されているファイルの正確なコピーとして復元するには、そのコンフィギュレーションファイルをスタートアップコンフィギュレーションに直接コピーし（**copy ftp:**| **rcp:**| **tftp:**} **nvram:startup-config** コマンドを使用）、デバイスをリロードする必要があります。

サーバからデバイスへコンフィギュレーションファイルをコピーするには、次の項で説明する作業を実行します。

使用するプロトコルは、使用中のサーバのタイプに応じて異なります。FTP および rcp のトランスポート メカニズムは、TFTP よりも高速でデータ配信の信頼性も優れています。これらの改善は、FTP および rcp のトランスポート メカニズムがコネクション型の TCP/IP スタック上に構築されており、これを使用しているために可能になりました。

Deviceから TFTP サーバへのコンフィギュレーション ファイルのコピー

一部の TFTP 実装では、TFTP サーバ上にダミー ファイルを作成し、読み取り、書き込み、および実行を許可してから、ダミーファイルを上書きする形でファイルをコピーする必要があります。詳細については、ご使用の TFTP のマニュアルを参照してください。

デバイスから RCP サーバへのコンフィギュレーション ファイルのコピー

デバイスから RCP サーバへコンフィギュレーション ファイルをコピーできます。

ネットワークを UNIX コミュニティでリソースとして使用する最初の試みの 1 つは、リモート シェル（RSH）およびリモート コピー（rcp）機能が含まれた、リモート シェル プロトコルの設計および実装につながりました。rsh および rcp により、ユーザはリモートでコマンドを実行し、ネットワーク上のリモート ホストまたはサーバにあるファイル システムからまたはファイル システムへファイルをコピーすることが可能になります。シスコの rsh および rcp 実装は、標準実装と相互運用できます。

rcp の **copy** コマンドは、リモート システム上の rsh サーバ（またはデーモン）に依存します。rcp を使用してファイルをコピーするために、TFTP のようにファイル配布用のサーバを作成する必要はありません。必要なのは、リモート シェル（rsh）をサポートするサーバへのアクセスだけです（ほとんどの UNIX システムは rsh をサポートしています）。ファイルをある場所から別の場所へコピーするため、コピー元ファイルに対する読み取り権限と、コピー先ファイルに対する書き込み権限が必要です。コピー先ファイルが存在しない場合は、rcp により作成されます。

シスコの rcp 実装は UNIX の rcp 実装（ネットワーク上のシステム間でファイルをコピー）の関数をエミュレートしたものですが、シスコのコマンド構文は UNIX の rcp コマンド構文とは異なります。シスコの rcp サポートは、rcp をトランスポート メカニズムとして使用する一連の **copy** コマンドを提供しています。これらの rcp **copy** コマンドは、シスコの TFTP **copy** コマンドに類似していますが、高速で信頼性の高いデータ配信を実現する代替方法を備えているという点が異なります。これらの改善は、rcp のトランスポート メカニズムがコネクション型の TCP/IP スタック上に構築されており、これを使用しているために可能になりました。rcp コマンドを使用して、デバイスからネットワークサーバ、またはその逆へシステムイメージおよびコンフィギュレーション ファイルをコピーできます。

また、rcp サポートをイネーブルにし、リモート システムのユーザがデバイスから、またはその逆へファイルをコピーできるようにすることも可能です。

リモート ユーザによるデバイスとのファイルのコピーができるように Cisco IOS ソフトウェアを設定するには、**iprcmdrcp-enable** グローバル コンフィギュレーション コマンドを使用します。

制限事項

RCP プロトコルでは、クライアントは RCP 要求ごとにリモート ユーザ名をサーバに送信する必要があります。RCP を使用してデバイスからサーバへコンフィギュレーション ファイルをコピーする場合、Cisco IOS ソフトウェアによって、次の順番で最初に発見された有効なユーザ名が送信されます。

1. **copy** EXEC コマンドで指定されたユーザ名（ユーザ名が指定されている場合）。
2. **ip rcmd remote-username** グローバル コンフィギュレーション コマンドで設定されたユーザ名（コマンドが設定されている場合）。
3. 現在の TTY（端末）プロセスに関連付けられているリモートユーザ名。たとえば、ユーザが Telnet を介してデバイスに接続されており、**username** コマンドを介して認証された場

合は、Telnetユーザ名がリモートユーザ名としてデバイス ソフトウェアによって送信されます。

4. デバイスの管理ホスト名。

RCP コピー要求を正常に実行するためには、ネットワーク サーバ上にリモート ユーザ名のアカウントを定義する必要があります。このサーバがディレクトリ構造をとっている場合、コンフィギュレーション ファイルまたはイメージは、サーバ上のリモート ユーザ名と関連付けられたディレクトリに書き込まれるか、そのディレクトリからコピーされます。たとえば、システム イメージがサーバ上のユーザのホーム ディレクトリにある場合は、そのユーザの名前をリモート ユーザ名として指定できます。

ip rcmd remote-username コマンドを使用して、すべてのコピーに対してユーザ名を指定します。（rcmd は、スーパーユーザ レベルで使用される UNIX ルーチンで、予約されたポート番号に基づいた認証スキームを使用してリモート マシン上でコマンドを実行します。rcmd は「Remote Command（リモート コマンド）」の略です）。特定のコピー操作にのみ使用するユーザ名を指定する場合は、**copy** コマンド内でユーザ名を指定します。

サーバに書き込む場合、デバイス上のユーザからの RCP 書き込み要求を受け入れるよう、RCP サーバを適切に設定する必要があります。UNIX システムの場合は、RCP サーバ上のリモート ユーザ用の .rhosts ファイルにエントリを追加する必要があります。たとえば、デバイスに次の設定行が含まれているとします。

```
hostname Device1
ip rcmd remote-username User0
```

デバイスの IP アドレスが デバイス1.example.com に変換される場合、RCP サーバ上の User0 の .rhosts ファイルには、次の行が含まれることになります。

```
Device1.example.com Device1
```

RCP ユーザ名に関する要件

RCP プロトコルでは、クライアントは RCP 要求ごとにリモート ユーザ名をサーバに送信する必要があります。RCP を使用してデバイスからサーバへコンフィギュレーション ファイルをコピーする場合、Cisco IOS ソフトウェアによって、次の順番で最初に発見された有効なユーザ名が送信されます。

1. **copy** EXEC コマンドで指定されたユーザ名（ユーザ名が指定されている場合）。
2. **ip rcmd remote-username** グローバル コンフィギュレーション コマンドで設定されたユーザ名（コマンドが設定されている場合）。
3. 現在の TTY（端末）プロセスに関連付けられているリモートユーザ名。たとえば、ユーザが Telnet を介してデバイスに接続されており、**username** コマンドを介して認証された場合は、Telnetユーザ名がリモートユーザ名としてデバイス ソフトウェアによって送信されます。
4. デバイスの管理ホスト名。

RCP コピー要求を実行するためには、ネットワーク サーバ上にリモート ユーザ名のアカウントを定義する必要があります。このサーバがディレクトリ構造をとっている場合、コンフィギュレーション ファイルまたはイメージは、サーバ上のリモート ユーザ名と関連付けられたディレクトリに書き込まれるか、そのディレクトリからコピーされます。たとえば、システム イメージがサーバ上のユーザのホームディレクトリにある場合は、そのユーザの名前をリモート ユーザ名として指定します。

詳細については、ご使用の RCP サーバのマニュアルを参照してください。

デバイスから FTP サーバへのコンフィギュレーション ファイルのコピー

デバイスから FTP サーバへコンフィギュレーション ファイルをコピーできます。

FTP ユーザ名およびパスワードの概要

FTPプロトコルでは、FTP要求ごとにリモートユーザ名およびパスワードを、クライアントがサーバに送信する必要があります。FTP を使用してデバイスからサーバへコンフィギュレーション ファイルをコピーする場合、Cisco IOS ソフトウェアは、次の順番で最初に発見した有効なユーザ名を送信します。

1. **copy** EXEC コマンドで指定されたユーザ名（ユーザ名が指定されている場合）。
2. **ip ftp username** グローバル コンフィギュレーション コマンドで設定されたユーザ名（コマンドが設定されている場合）。
3. Anonymous

デバイスは次の順番で最初に発見した有効なパスワードを送信します。

1. **copy** コマンドで指定されたパスワード（パスワードが指定されている場合）。
2. **ip ftp password** コマンドで設定されたパスワード（コマンドが設定されている場合）。
3. デバイスは、*username*@デバイス*name.domain* というパスワードを生成します。変数 *username* は現在のセッションと関連付けられたユーザ名、デバイス *name* は設定済みホスト名、*domain* はデバイスのドメインです。

ユーザ名およびパスワードは、FTP サーバのアカウントに関連付けられている必要があります。サーバに書き込む場合、デバイス上のユーザからのFTP書き込み要求を受け入れるよう、FTP サーバを適切に設定する必要があります。

このサーバがディレクトリ構造をとっている場合、コンフィギュレーションファイルまたはイメージは、サーバ上のユーザ名と関連付けられたディレクトリに書き込まれるか、そのディレクトリからコピーされます。たとえば、システム イメージがサーバ上のユーザのホーム ディレクトリにある場合は、そのユーザの名前をリモート ユーザ名として指定します。

詳細については、ご使用の FTP サーバのマニュアルを参照してください。

ip ftp username および **ip ftp password** グローバル コンフィギュレーション コマンドを使用して、すべてのコピーに対してユーザ名とパスワードを指定します。当該のコピー操作だけに対してユーザ名を指定する場合は、**copy** EXEC コマンドにユーザ名を含めます。

VRF によるファイルのコピー

copy コマンドで指定した VRF インターフェイス経由でファイルをコピーできます。設定の変更リクエストを使用せずに直接送信元インターフェイスを変更できるので、**copy** コマンドで VRF を指定するほうが簡単で効率的です。

例

次の例に、**copy** コマンドを使用して、VRF 経由でファイルをコピーする方法を示します。

```
Device# copy scp: flash-1: vrf test-vrf
Address or name of remote host [10.1.2.3]?
Source username [ScpUser]?
Source filename [/auto/tftp-server/ScpUser/vrf_test.txt]?
Destination filename [vrf_test.txt]?
Getting the vrf name as test-vrf
Password:
Sending file modes: C0644 10 vrf_test.txt
!
223 bytes copied in 22.740 secs (10 bytes/sec)
```

スイッチから別のスイッチへのコンフィギュレーション ファイルのコピー

あるスイッチから別のスイッチに設定をコピーすることができます。これは2ステッププロセスです。スイッチから TFTP サーバに設定をコピーし、次に TFTP から別のスイッチに設定をコピーします。

スイッチから現在の設定をコピーするには、**copy startup-config tftp:** コマンドを実行し、続く指示に従います。設定が TFTP サーバにコピーされます。

次に、別のスイッチへログインし、**copy tftp: startup-config** コマンドを実行して、続く指示に従います。これで、設定は別のスイッチにコピーされます。

設定をコピーした後、その設定を保存するには、**write memory** コマンドを使用し、その後スイッチをリロードするか、または **copy startup-config running-config** コマンドを実行します。

NVRAM より大きいコンフィギュレーション ファイル

NVRAM より大きいコンフィギュレーションファイルを維持管理するには、以降の項の情報を知っておく必要があります。

コンフィギュレーション ファイルの圧縮

service compress-config グローバル コンフィギュレーション コマンドは、コンフィギュレーション ファイルを圧縮して NVRAM に格納することを指定します。コンフィギュレーションファイルが圧縮されると、デバイスは正常に機能します。システムの起動時に、システムはコンフィギュレーションファイルが圧縮されていることを認識し、圧縮されたコンフィギュレーション ファイルを展開して、正常に処理を進めます。**more nvram:startup-config** EXEC コマンドにより、コンフィギュレーションが展開されてから表示されます。

コンフィギュレーションファイルを圧縮する前に、適切なハードウェアのインストレーションおよびメンテナンス マニュアルを参照してください。ご利用のシステムの ROM がファイル圧縮をサポートしていることを確認します。サポートしていない場合、ファイル圧縮をサポートしている新しい ROM をインストールできます。

コンフィギュレーションのサイズは、NVRAM のサイズの 3 倍を超えてはいけません。NVRAM のサイズが 128 KB の場合、展開できる最大のコンフィギュレーション ファイルのサイズは 384 KB です。

service compress-config グローバル コンフィギュレーション コマンドは、Cisco IOS ソフトウェア リリース 10.0 以降のブート ROM を使用している場合に限り実行できます。新しい ROM をインストールするのは 1 回限りの操作で、ROM に Cisco IOS Release 10.0 がない場合だけ必要です。ブート ROM が圧縮コンフィギュレーションを認識しない場合は、次のメッセージが表示されます。

```
Boot ROMs do not support NVRAM compression Config NOT written to NVRAM
```

コンフィギュレーションのクラス A フラッシュ ファイル システム上のフラッシュ メモリへの格納

クラス A フラッシュ ファイル システムのデバイス上では、内部フラッシュ メモリのファイルまたは PCMCIA スロットのフラッシュ メモリのファイルに **CONFIG_FILE** 環境変数を設定することにより、スタートアップ コンフィギュレーションをフラッシュ メモリに格納できます。

詳細については、「クラス A フラッシュ ファイル システムでの CONFIG_FILE 環境変数の指定 （197 ページ）」を参照してください。

大きいコンフィギュレーションを編集または変更する場合は、注意する必要があります。フラッシュ メモリ領域は **copy system:running-config nvram:startup-config** EXEC コマンドが発行されるたびに使用されます。フラッシュ メモリのファイル管理（空き領域の最適化などの）は自動的には行われないため、利用可能なフラッシュ メモリに十分注意を払う必要があります。**squeeze** コマンドを使用して、使用済み領域を再要求します。20 MB 以上の大容量フラッシュ カードを使用することを推奨します。

ネットワークからのコンフィギュレーション コマンドのロード

コンフィギュレーションが大きい場合は、FTP、RCP、TFTP のいずれかのサーバに格納しておき、システムの起動時にダウンロードすることもできます。ネットワーク サーバを使用して大規模な設定を格納するには、Device から TFTP サーバへのコンフィギュレーション ファイルのコピー （175 ページ）およびコンフィギュレーション ファイルをダウンロードするデバイスの設定 （171 ページ）の項でこれらのコマンドの詳細を参照してください。

コンフィギュレーション ファイルをダウンロードするデバイスの設定

システムの起動時に 1 つまたは 2 つのコンフィギュレーション ファイルをロードするようにデバイスを設定できます。コンフィギュレーション ファイルは、コマンドラインにコマンドを入力した場合と同様に、メモリにロードされ読み込まれます。そのため、デバイスのコンフィギュレーションは、元のスタートアップ コンフィギュレーションと 1 つまたは 2 つのダウンロードされたコンフィギュレーション ファイルが混在したものになります。

ネットワークとホストのコンフィギュレーション ファイル

歴史的な理由から、デバイスが最初にダウンロードするファイルは、ネットワーク コンフィギュレーション ファイルと呼ばれます。デバイスが2番目にダウンロードするファイルは、ホスト コンフィギュレーション ファイルと呼ばれます。2 つのコンフィギュレーション ファイルは、ネットワーク上のすべてのデバイスが同一コマンドの多くを使用する場合に使用できます。ネットワーク コンフィギュレーション ファイルには、すべてのデバイスを設定するために使用される標準コマンドが含まれます。ホスト コンフィギュレーション ファイルには、特定の 1 つのホストに固有のコマンドが含まれます。2 つのコンフィギュレーション ファイルをロードする場合、ホスト コンフィギュレーション ファイルを、もう 1 つのファイルより優先させる必要があります。ネットワーク コンフィギュレーション ファイルとホスト コンフィギュレーション ファイルの両方とも、TFTP、RCP、FTP のいずれかを介して到達可能なネットワーク サーバ上にあり、読み取り可能である必要があります。

コンフィギュレーション ファイル情報の管理方法

コンフィギュレーション ファイル情報の表示

コンフィギュレーション ファイルに関する情報を表示するには、このセクションの手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**showboot** 例： `Device# show boot`	BOOT 環境変数の内容（設定されている場合）、CONFIG_FILE 環境変数によって指定されているコンフィギュレーション ファイルの名前、および BOOTLDR 環境変数の内容を示します。
ステップ 3	**more** *file-url* 例： `Device# more 10.1.1.1`	指定されたファイルの内容を表示します。
ステップ 4	**showrunning-config** 例：	実行コンフィギュレーション ファイルの内容を表示します（**more**

	コマンドまたはアクション	目的
	`Device# show running-config`	**system:running-config** コマンドのコマンド エイリアス）。
ステップ 5	**showstartup-config** 例： `Device# show startup-config`	スタートアップ コンフィギュレーション ファイルの内容を表示します。（**more nvram:startup-config** コマンドのコマンド エイリアス）。 クラス A フラッシュ ファイル システム プラットフォーム以外のすべてのプラットフォーム上では、通常、デフォルトの startup-config ファイルは NVRAM に格納されます。 クラス A フラッシュ ファイル システム プラットフォーム上では、CONFIG_FILE 環境変数はデフォルトの startup-config ファイルを指定します。 CONFIG_FILE 変数のデフォルトは NVRAM になります。

コンフィギュレーション ファイルの変更

Cisco IOS ソフトウェアは、1 行につき 1 つのコンフィギュレーション コマンドを受け入れます。コンフィギュレーションコマンドは、必要なだけ入力できます。コンフィギュレーションファイルには、入力したコマンドを説明するコメントを追加できます。コメントの先頭には、感嘆符（!）を付けます。コメントは NVRAM にもコンフィギュレーション ファイルのアクティブ コピーにも格納されないため、**show running-config** または **moresystem:running-config** EXEC コマンドでアクティブな設定のリストを表示しても、コメントは表示されません。また、**show startup-config** または **more nvram:startup-config** EXEC モード コマンドでスタートアップ コンフィギュレーションのリストを表示しても、コメントは表示されません。コメントは、コンフィギュレーションファイルがデバイスにロードされたときにコンフィギュレーション ファイルから削除されます。ただし、ファイル転送プロトコル（FTP）、リモート コピー プロトコル（RCP）、または Trivial File Transfer Protocol（TFTP）サーバ上に格納されているコンフィギュレーションファイルのコメントのリストは表示できます。CLI を使用してソフトウェアは設定するときは、ユーザの入力に従ってソフトウェアによりコマンドが実行されます。CLI を使用してソフトウェアを設定するには、特権 EXEC モードを開始して次のコマンドを使用します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	**configurationcommand** 例： `Device(config)# configuration command`	必要なコンフィギュレーション コマンドを入力します。Cisco IOS マニュアル セットに、テクノロジー別に編成されたコンフィギュレーション コマンドが説明されています。
ステップ 4	次のいずれかを実行します。 • **end** • **^Z** 例： `Device(config)# end`	コンフィギュレーション セッションを終了し、EXEC モードに戻ります。 （注） Ctrl キーと Z キーを同時に押すと、画面に ^Z と表示されます。
ステップ 5	**copysystem:running-confignvram:startup-config** 例： `Device# copy system:running-config nvram:startup-config`	実行コンフィギュレーション ファイルをスタートアップ コンフィギュレーション ファイルとして保存します。 **copy running-config startup-config** コマンド エイリアスも使用できますが、このコマンドは精度が高くないため、注意する必要があります。ほとんどのプラットフォーム上では、このコマンドによりコンフィギュレーションは NVRAM に保存されます。クラス A フラッシュ ファイル システムのプラットフォーム上では、この手順によりコンフィギュレーションは CONFIG_FILE 環境変数によって指定された場所に保存されます（デフォルトの CONFIG_FILE 変数では、ファイルの保存先は NVRAM に指定されています）。

例

次の例では、デバイスのデバイスプロンプト名が設定されています。感嘆符（!）で示されたコメント行では、いずれのコマンドも実行されません。**hostname** コマンドは、デバイスから new_name へデバイス名を変更するために使用されます。Ctrl-Z（^Z）キーを押すか、**end** コマンドを入力すると、コンフィギュレーション モードが終了します。**copy system:running-config nvram:startup-config** コマンドにより、現在の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ます。

```
Device# configure terminal
Device(config)# !The following command provides the switch host name.
Device(config)# hostname new_name
new_name(config)# end
new_name# copy system:running-config nvram:startup-config
```

スタートアップ コンフィギュレーションが NVRAM にある場合は、現在の設定情報がコンフィギュレーション コマンドとしてテキスト形式で格納され、デフォルト以外の設定だけが記録されます。破損データから保護するために、メモリはチェックサム算出されます。

（注） 一部の特定のコマンドは、NVRAM に保存されない場合があります。これらのコマンドは、マシンをリブートしたときに再入力する必要があります。これらのコマンドは、マニュアルに記載されています。リブート後にすばやくデバイスを再設定できるように、これらの設定のリストを保管しておくことを推奨します。

DeviceからTFTPサーバへのコンフィギュレーションファイルのコピー

TFTP ネットワーク サーバ上の設定をコピーするには、以下の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**copysystem:running-configtftp:**[[[*//location*]*/directory*]*/filename*] 例：	TFTP サーバへ実行コンフィギュレーション ファイルをコピーします。

	コマンドまたはアクション	目的
	`Device# copy system:running-config tftp: //server1/topdir/file10`	
ステップ 3	**copynvram:startup-configtftp:**[[[*//location]/directory]/filename*] 例： `Device# copy nvram:startup-config tftp: //server1/1stdir/file10`	TFTP サーバへスタートアップ コンフィギュレーション ファイルをコピーします。

例

次に、デバイスから TFTP サーバへコンフィギュレーション ファイルをコピーする例を示します。

```
Device# copy system:running-config tftp://172.16.2.155/tokyo-confg
Write file tokyo-confg on host 172.16.2.155? [confirm] Y
Writing tokyo-confg!!! [OK]
```

次の作業

copy コマンドを発行した後、追加情報またはアクションの確認を求めるプロンプトが表示される場合があります。表示されるプロンプトは、**copy** コマンドで入力した情報量および **file prompt** グローバル コンフィギュレーション コマンドの現在の設定によって異なります。

Deviceから RCP サーバへのコンフィギュレーション ファイルのコピー

デバイスから RCP サーバへスタートアップ コンフィギュレーション ファイルまたは実行コンフィギュレーション ファイルをコピーするには、特権 EXEC モードを開始して次のコマンドを使用します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	**ip rcmd remote-username** ユーザ名 例： `Device(config)# ip rcmd remote-username NetAdmin1`	（任意）デフォルトのリモート ユーザ名を変更します。
ステップ 4	**end** 例： `Device(config)# end`	（任意）グローバル コンフィギュレーション モードを終了します。

ステップ 5	次のいずれかを実行します。
	• **copy system:running-config rcp:**[[[//[*username*@]*location*]/*directory*]/*filename*]
	• **copy nvram:startup-config rcp:**[[[//[*username*@]*location*]/*directory*]/*filename*]
	例：
	`Device# copy system:running-config rcp:`
	`//NetAdmin1@example.com/dir-files/file1`
		• デバイスの実行コンフィギュレーション ファイルが RCP サーバ上に格納されるように指定します。
		または
		• デバイスのスタートアップ コンフィギュレーション ファイルが RCP サーバ上に格納されるように指定します。

例

RCP サーバへの実行コンフィギュレーション ファイルの格納

次に、rtr2-confg という名前の実行コンフィギュレーションファイルを IP アドレス 172.16.101.101 のリモート ホスト上の netadmin1 ディレクトリにコピーする例を示します。

```
Device# copy system:running-config rcp://netadmin1@172.16.101.101/runfile2-confg
Write file runfile2-confg on host 172.16.101.101?[confirm]
Building configuration...[OK]
Connected to 172.16.101.101
Device#
```

RCP サーバへのスタートアップ コンフィギュレーション ファイルの格納

次に、RCP を使用してファイルをコピーすることによって、サーバ上にスタートアップ コンフィギュレーション ファイルを格納する例を示します。

```
Device# configure terminal

Device(config)# ip rcmd remote-username netadmin2

Device(config)# end

Device# copy nvram:startup-config rcp:
```

```
Remote host[]? 172.16.101.101

Name of configuration file to write [start-confg]?
Write file start-confg on host 172.16.101.101?[confirm]
![OK]
```

次の作業

copy EXEC コマンドを発行した後、追加情報またはアクションの確認を求めるプロンプトが表示される場合があります。表示されるプロンプトは、**copy** コマンドで入力した情報量および **file prompt** グローバル コンフィギュレーション コマンドの現在の設定によって異なります。

デバイスから FTP サーバへのコンフィギュレーション ファイルのコピー

デバイスから FTP サーバへスタートアップ コンフィギュレーション ファイルまたは実行コンフィギュレーション ファイルをコピーするには、以下の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	デバイス上で、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 3	**ipftpusername** ユーザ名 例： `Device(config)# ip ftp username NetAdmin1`	（任意）デフォルトのリモート ユーザ名を指定します。
ステップ 4	**ipftppassword** *password* 例： `Device(config)# ip ftp password adminpassword`	（任意）デフォルトのパスワードを指定します。
ステップ 5	**end** 例：	（任意）グローバル コンフィギュレーション モードを終了します。この手順は、デフォルトのリモート ユーザ名ま

	コマンドまたはアクション	目的
	`Device(config)# end`	たはパスワードを上書きする場合にだけ必要です（ステップ 2 および 3 を参照）。
ステップ 6	次のいずれかを実行します。 • **copysystem:running-configftp:**[[[//[*username* [**:***password*]**@**]*location*]/*directory*]/*filename*] or • **copynvram:startup-config ftp:**[[[//[*username* [**:***password*]**@**]*location*]/*directory*]/*filename*] 例： `Device# copy system:running-config ftp:`	FTP サーバの指定された場所へ実行コンフィギュレーションまたはスタートアップ コンフィギュレーション ファイルをコピーします。

例

FTP サーバへの実行コンフィギュレーション ファイルの格納

次に、runfile-confg という名前の実行コンフィギュレーション ファイルを IP アドレス 172.16.101.101 のリモート ホスト上の netadmin1 ディレクトリにコピーする例を示します。

```
Device# copy system:running-config ftp://netadmin1:mypass@172.16.101.101/runfile-confg
Write file runfile-confg on host 172.16.101.101?[confirm]
Building configuration...[OK]
Connected to 172.16.101.101
Device#
```

FTP サーバへのスタートアップ コンフィギュレーション ファイルの格納

次に、FTP を使用してファイルをコピーすることによって、サーバ上にスタートアップ コンフィギュレーション ファイルを格納する例を示します。

```
Device# configure terminal

Device(config)# ip ftp username netadmin2

Device(config)# ip ftp password mypass

Device(config)# end

Device# copy nvram:startup-config ftp:

Remote host[]? 172.16.101.101

Name of configuration file to write [start-confg]?
Write file start-confg on host 172.16.101.101?[confirm]
![OK]
```

次の作業

copy EXEC コマンドを発行した後、追加情報またはアクションの確認を求めるプロンプトが表示される場合があります。表示されるプロンプトは、**copy** コマンドで入力した情報量および **file prompt** グローバル コンフィギュレーション コマンドの現在の設定によって異なります。

TFTP サーバからデバイスへのコンフィギュレーション ファイルのコピー

TFTP サーバからデバイスへコンフィギュレーション ファイルをコピーするには、以下の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**copytftp:**[[[*//location*]/*directory*]/*filename*] **system:running-config** 例： `Device# copy tftp://server1/dir10/datasource system:running-config`	TFTP サーバから実行コンフィギュレーションへコンフィギュレーション ファイルをコピーします。
ステップ 3	**copytftp:**[[[*//location*]/*directory*]/*filename*] **nvram:startup-config** 例： `Device# copy tftp://server1/dir10/datasource nvram:startup-config`	TFTP サーバからスタートアップ コンフィギュレーションへコンフィギュレーション ファイルをコピーします。
ステップ 4	**copytftp:**[[[*//location*]/*directory*]/*filename*] **flash:**[*partition*]/[*directory*]/**startup-config** 例： `Device# copy tftp://server1/dir10/datasource flash:startup-config`	TFTP サーバからスタートアップ コンフィギュレーションへコンフィギュレーション ファイルをコピーします。

例

次に、IP アドレス 172.16.2.155 にある、**tokyo-confg**という名前のファイルからソフトウェアを設定する例を示します。

```
Device# copy tftp://172.16.2.155/tokyo-confg system:running-config

Configure using tokyo-confg from 172.16.2.155? [confirm] Y

Booting tokyo-confg from 172.16.2.155:!!! [OK - 874/16000 bytes]
```

次の作業

copy EXEC コマンドを発行した後、追加情報またはアクションの確認を求めるプロンプトが表示される場合があります。表示されるプロンプトは、**copy** コマンドで入力した情報量および **file prompt** グローバル コンフィギュレーション コマンドの現在の設定によって異なります。

rcpサーバからデバイスへのコンフィギュレーションファイルのコピー

rcp サーバから実行コンフィギュレーションまたはスタートアップ コンフィギュレーションへコンフィギュレーション ファイルをコピーするには、以下の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	（任意）端末からコンフィギュレーション モードを開始します。この手順は、デフォルトのリモート ユーザ名を上書きする場合にだけ必要です（ステップ 3 を参照）。
ステップ 3	**iprcmdremote-username** ユーザ名 例： `Device(config)# ip rcmd remote-username NetAdmin1`	（任意）リモート ユーザ名を指定します。
ステップ 4	**end** 例：	（任意）グローバル コンフィギュレーション モードを終了します。この手順は、デフォルトのリモート ユーザ名ま

	コマンドまたはアクション	目的
	`Device(config)# end`	たはパスワードを上書きする場合にだけ必要です（ステップ 2 を参照）。
ステップ 5	次のいずれかを実行します。 • **copy rcp:[[[//[*username*@]*location*]/*directory*]/*filename*] system:running-config** • **copy rcp:[[[//[*username*@]*location*]/*directory*]/*filename*] nvram:startup-config** 例： `Device# copy rcp://[user1@example.com/dir10/fileone] nvram:startup-config`	rcp サーバから実行コンフィギュレーションまたはスタートアップ コンフィギュレーションへコンフィギュレーション ファイルをコピーします。

例

rcp の Running-Config のコピー

次に、host1-confg という名前のコンフィギュレーション ファイルを、IP アドレスが 172.16.101.101 のリモート サーバ上の netadmin1 ディレクトリからコピーし、デバイスでコマンドをロードし実行する例を示します。

```
Device# copy rcp://netadmin1@172.16.101.101/host1-confg system:running-config
Configure using host1-confg from 172.16.101.101? [confirm]
Connected to 172.16.101.101
Loading 1112 byte file host1-confg:![OK]
Device#
%SYS-5-CONFIG: Configured from host1-config by rcp from 172.16.101.101
```

rcp の Startup-Config のコピー

次に、リモートユーザ名 netadmin1 を指定する例を示します。次に host2-confg という名前のコンフィギュレーション ファイルを、IP アドレスが 172.16.101.101 のリモート サーバ上の netadmin1 ディレクトリからスタートアップ コンフィギュレーションへコピーします。

```
Device# configure terminal
Device(config)# ip rcmd remote-username netadmin1
Device(config)# end
Device# copy rcp: nvram:startup-config
Address of remote host [255.255.255.255]? 172.16.101.101
Name of configuration file[rtr2-confg]? host2-confg
Configure using host2-confg from 172.16.101.101?[confirm]
Connected to 172.16.101.101
Loading 1112 byte file host2-confg:![OK]
[OK]
Device#
%SYS-5-CONFIG_NV:Non-volatile store configured from host2-config by rcp from 172.16.101.101
```

次の作業

copy EXEC コマンドを発行した後、追加情報またはアクションの確認を求めるプロンプトが表示される場合があります。表示されるプロンプトは、**copy** コマンドで入力した情報量および **file prompt** グローバル コンフィギュレーション コマンドの現在の設定によって異なります。

FTP サーバからデバイスへのコンフィギュレーション ファイルのコピー

FTP サーバから実行コンフィギュレーションまたはスタートアップ コンフィギュレーションへコンフィギュレーション ファイルをコピーするには、以下の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	（任意）グローバル コンフィギュレーション モードを開始できます。この手順は、デフォルトのリモート ユーザ名またはパスワードを上書きする場合にだけ必要です（ステップ 3 および 4 を参照）。
ステップ 3	**ipftpusername** ユーザ名 例： `Device(config)# ip ftp username NetAdmin1`	（任意）デフォルトのリモート ユーザ名を指定します。
ステップ 4	**ipftppassword** *password* 例： `Device(config)# ip ftp password adminpassword`	（任意）デフォルトのパスワードを指定します。
ステップ 5	**end** 例： `Device(config)# end`	（任意）グローバル コンフィギュレーション モードを終了します。この手順は、デフォルトのリモート ユーザ名またはパスワードを上書きする場合にだけ必要です（ステップ 3 および 4 を参照）。

	コマンドまたはアクション	目的
ステップ 6	次のいずれかを実行します。 • **copyftp:** [[[//[*username*[**:***password*]**@**]*location*] /*directory*]/*filename*]**system:running-config** • **copyftp:** [[[//[*username*[**:***password*]**@**]*location*] /*directory*]/*filename*]**nvram:startup-config** 例： `Device# copy ftp:nvram:startup-config`	FTP を使用して、ネットワーク サーバから実行メモリまたはスタートアップ コンフィギュレーションへコンフィギュレーション ファイルをコピーします。

例

FTP の Running-Config のコピー

次に、host1-confg という名前のコンフィギュレーション ファイルを、IP アドレスが 172.16.101.101 のリモート サーバ上の netadmin1 ディレクトリからコピーし、デバイスでコマンドをロードし実行する例を示します。

```
Device# copy ftp://netadmin1:mypass@172.16.101.101/host1-confg system:running-config
Configure using host1-confg from 172.16.101.101? [confirm]
Connected to 172.16.101.101
Loading 1112 byte file host1-confg:![OK]
Device#
%SYS-5-CONFIG: Configured from host1-config by ftp from 172.16.101.101
```

FTP の Startup-Config のコピー

次に、リモート ユーザ名 netadmin1 を指定する例を示します。次に host2-confg という名前のコンフィギュレーション ファイルを、IP アドレスが 172.16.101.101 のリモート サーバ上の netadmin1 ディレクトリからスタートアップ コンフィギュレーションへコピーします。

```
Device# configure terminal
Device(config)# ip ftp username netadmin1
Device(config)# ip ftp password mypass
Device(config)# end
Device# copy ftp: nvram:startup-config
Address of remote host [255.255.255.255]? 172.16.101.101
Name of configuration file[host1-confg]? host2-confg
Configure using host2-confg from 172.16.101.101?[confirm]
Connected to 172.16.101.101
Loading 1112 byte file host2-confg:![OK]
[OK]
Device#
%SYS-5-CONFIG_NV:Non-volatile store configured from host2-config by ftp from 172.16.101.101
```

次の作業

copy EXEC コマンドを発行した後、追加情報またはアクションの確認を求めるプロンプトが表示される場合があります。表示されるプロンプトは、**copy** コマンドで入力した情報量および **file prompt** グローバル コンフィギュレーション コマンドの現在の設定によって異なります。

NVRAM より大きいコンフィギュレーション ファイルの保守

NVRAM のサイズを超えるコンフィギュレーション ファイルを保守するには、以降のセクションで説明するタスクを実行します。

コンフィギュレーション ファイルの圧縮

コンフィギュレーション ファイルを圧縮するには、このセクションの手順を実行してください。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	**servicecompress-config** 例： `Device(config)# service compress-config`	コンフィギュレーション ファイルを圧縮することを指定します。
ステップ 4	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了します。
ステップ 5	次のいずれかを実行します。 • 新しいコンフィギュレーションをコピーするには、FTP、RCP、TFTP を使用します。	新しいコンフィギュレーションを入力します。 • NVRAM のサイズの 3 倍以上のコンフィギュレーションをロードしよう

	コマンドまたはアクション	目的
	• **configureterminal** 例： `Device# configure terminal`	とすると、次のエラー メッセージが表示されます。 「[buffer overflow - *file-size* /*buffer-size* bytes]。」
ステップ 6	**copysystem:running-confignvram:startup-config** 例： `Device(config)# copy system:running-config nvram:startup-config`	実行コンフィギュレーションの変更が終わったら、新しいコンフィギュレーションを保存します。

例

次に、129 KB のコンフィギュレーション ファイルを 11 KB に圧縮する例を示します。

```
Device# configure terminal

Device(config)# service compress-config

Device(config)# end

Device# copy tftp://172.16.2.15/tokyo-confg system:running-config

Configure using tokyo-confg from 172.16.2.155? [confirm] y

Booting tokyo-confg from 172.16.2.155:!!! [OK - 874/16000 bytes]
Device# copy system:running-config nvram:startup-config

Building configuration...
Compressing configuration from 129648 bytes to 11077 bytes
[OK]
```

コンフィギュレーションのクラス A フラッシュ ファイル システム上のフラッシュ メモリへの格納

スタートアップ コンフィギュレーションをフラッシュ メモリに格納するには、このセクションの手順を実行してください。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。

	コマンドまたはアクション	目的
ステップ 2	**copynvram:startup-config** *flash-filesystem:filename* 例： `Device# copy nvram:startup-config usbflash0:switch-config`	新しい場所に現在のスタートアップ コンフィギュレーションをコピーして、コンフィギュレーション ファイルを作成します。
ステップ 3	**configureterminal** 例： `Device# configure terminal`	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 4	**bootconfigflash-filesystem:** *filename* 例： `Device(config)# boot config usbflash0:switch-config`	CONFIG_FILE 環境変数を設定することにより、フラッシュ メモリにスタートアップ コンフィギュレーション ファイルを格納することを指定します。
ステップ 5	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了します。
ステップ 6	次のいずれかを実行します。 • 新しいコンフィギュレーションをコピーするには、FTP、RCP、TFTP を使用します。NVRAM サイズの 3 倍を超える大きさのコンフィギュレーションをロードしようとすると、次のエラー メッセージが表示されます。「[buffer overflow - *file-size* /*buffer-size* bytes]」 • **configureterminal** 例： `Device# configure terminal`	新しいコンフィギュレーションを入力します。
ステップ 7	**copysystem:running-confignvram:startup-config** 例： `Device(config)# copy system:running-config nvram:startup-config`	実行コンフィギュレーションの変更が終わったら、新しいコンフィギュレーションを保存します。

例

以下に、usbflash0: に格納したコンフィギュレーションの例を示します。

```
Device# copy nvram:startup-config usbflash0:switch-config

Device# configure terminal

Device(config)# boot config usbflash0:switch-config

Device(config)# end

Device# copy system:running-config nvram:startup-config
```

ネットワークからのコンフィギュレーション コマンドのロード

ネットワーク サーバを使用して、大きなコンフィギュレーションを保存するには、このセクションの手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**copysystem:running-config** {**ftp:** | **rcp:** | **tftp:**} 例： `Device# copy system:running-config ftp:`	実行コンフィギュレーションを FTP、RCP、TFTP のいずれかのサーバに保存します。
ステップ 3	**configureterminal** 例： `Device# configure terminal`	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 4	**bootnetwork**{**ftp:**[[[//[*username* [**:***password*]**@**]*location*]/*directory*]/*filename*] | **rcp:**[[[//[*username***@**]*location*]/*directory*]/*filename*] | **tftp:**[[[//*location*]/*directory*]/*filename*]} 例： `Device(config)# boot network`	起動時にスタートアップ コンフィギュレーション ファイルをネットワーク サーバからロードすることを指定します。

	コマンドまたはアクション	目的
	`ftp://user1:guessme@example.com/dir10/file1`	
ステップ 5	**serviceconfig** 例： `Device(config)# service config`	システムの起動時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ようにスイッチをイネーブルにします。
ステップ 6	**end** 例： `Device(config)# end`	グローバル 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	**copysystem:running-confignvram:startup-config** 例： `Device# copy system:running-config nvram:startup-config`	設定を保存します。

フラッシュ メモリからスタートアップまたは実行コンフィギュレーションへのコンフィギュレーション ファイルのコピー

フラッシュ メモリから現在の NVRAM にあるスタートアップ コンフィギュレーションまたは実行コンフィギュレーションへコンフィギュレーション ファイルを直接コピーするには、ステップ 2 のいずれかのコマンドを入力します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	次のいずれかを実行します。 • **copy** *filesystem*: [*partition-number*:][*filename*] **nvram:startup-config** • **copy** *filesystem*: [*partition-number*:][*filename*] **system:running-config** 例：	• NVRAM にコンフィギュレーション ファイルを直接ロードする、または • 現在の実行コンフィギュレーションにコンフィギュレーション ファイルをコピーします。

	コマンドまたはアクション	目的
	`Device# copy usbflash0:4:ios-upgrade-1 nvram:startup-config`	

例

次に、usbflash0 にあるフラッシュ メモリ PC カードのパーティション 4 からデバイスのスタートアップ コンフィギュレーションへ ios-upgrade-1 という名前のファイルをコピーする例を示します。

```
Device# copy usbflash0:4:ios-upgrade-1 nvram:startup-config

Copy 'ios-upgrade-1' from flash device as 'startup-config' ? [yes/no] yes

[OK]
```

フラッシュ メモリ ファイル システム間でのコンフィギュレーション ファイルのコピー

複数のフラッシュ メモリ ファイル システムを備えたプラットフォーム上では、内部フラッシュ メモリなどのフラッシュ メモリ ファイル システムから他のフラッシュ メモリ ファイル システムへファイルをコピーできます。異なるフラッシュ メモリ ファイル システムへファイルをコピーすると、使用中のコンフィギュレーションのバックアップ コピーを作成し、他のデバイスにコンフィギュレーションを複製できます。フラッシュ メモリ ファイル システム間でコンフィギュレーション ファイルをコピーするには、EXEC モードで次のコマンドを使用します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Device> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**show** *source-filesystem*: **例：** `Device# show flash:`	フラッシュ メモリのレイアウトと内容を表示して、ファイル名を確認します。
ステップ 3	**copy** *source-filesystem*:[*partition-number*:][*filename*] *dest-filesystem*:[*partition-number*:][*filename*]	フラッシュ メモリ デバイス間でコンフィギュレーション ファイルをコピーします。

	コマンドまたはアクション	目的
	例： `Device# copy flash: usbflash0:`	- コピー元デバイスとコピー先デバイスは同じにはできません。例えば、**copy usbflash0: usbflash0:** コマンドは無効です。

例

次に、内部フラッシュ メモリのパーティション 1 からデバイス上の usbflash0 のパーティション 1 へ running-config という名前のファイルをコピーする例を示します。この例では、コピー元のパーティションが指定されていないため、デバイスからパーティション番号を要求されます。

```
Device# copy flash: usbflash0:

System flash
Partition   Size    Used      Free      Bank-Size  State          Copy Mode
  1         4096K   3070K     1025K     4096K      Read/Write     Direct
  2        16384K   1671K    14712K     8192K      Read/Write     Direct
[Type ?<no> for partition directory; ? for full directory; q to abort]
Which partition? [default = 1]
System flash directory, partition 1:
File  Length   Name/status
  1   3142748  dirt/network/mars-test/c3600-j-mz.latest
  2   850      running-config
[3143728 bytes used, 1050576 available, 4194304 total]
usbflash0 flash directory:
File  Length   Name/status
  1   1711088  dirt/gate/c3600-i-mz
  2   850      running-config
[1712068 bytes used, 2482236 available, 4194304 total]
Source file name? running-config

Destination file name [running-config]?
Verifying checksum for 'running-config' (file # 2)...  OK
Erase flash device before writing? [confirm]
Flash contains files. Are you sure you want to erase? [confirm]
Copy 'running-config' from flash: device
  as 'running-config' into usbflash0: device WITH erase? [yes/no] yes

Erasing device... eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee
...erased!
 [OK - 850/4194304 bytes]
Flash device copy took 00:00:30 [hh:mm:ss]
Verifying checksum...  OK (0x16)
```

FTP サーバからフラッシュ メモリ デバイスへのコンフィギュレーション ファイルのコピー

FTP サーバからフラッシュ メモリ デバイスへコンフィギュレーション ファイルをコピーするには、以下の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	（任意）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。この手順は、デフォルトのリモート ユーザ名またはパスワードを上書きする場合にだけ必要です（ステップ 3 および 4 を参照）。
ステップ 3	**ipftpusername** ユーザ名 例： `Device(config)# ip ftp username Admin01`	（任意）リモート ユーザ名を指定します。
ステップ 4	**ipftppassword** *password* 例： `Device(config)# ip ftp password adminpassword`	（任意）リモート パスワードを指定します。
ステップ 5	**end** 例： `Device(config)# end`	（任意）コンフィギュレーション モードを終了します。このステップが必要になるのは、デフォルトのリモート ユーザ名を上書きする場合のみです（ステップ 3 および 4 を参照）。
ステップ 6	**copyftp:**[[*//location*]/*directory*]/*bundle_name***flash:** 例： `Device>copy ftp://cat3k_caa-universalk9.SSA.03.12.02.EZP.150-12.02.EZP.150-12.02.EZP.bin flash:`	FTP を使用してネットワーク サーバからフラッシュ メモリ デバイスへコンフィギュレーション ファイルをコピーします。

次の作業

copy EXEC コマンドを発行した後、追加情報またはアクションの確認を求めるプロンプトが表示される場合があります。表示されるプロンプトは、**copy** コマンドで入力した情報量および **file prompt** グローバル コンフィギュレーション コマンドの現在の設定によって異なります。

RCPサーバからフラッシュ メモリ デバイスへのコンフィギュレーション ファイルのコピー

RCP サーバからフラッシュ メモリ デバイスへコンフィギュレーション ファイルをコピーするには、以下の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	（任意）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。この手順は、デフォルトのリモート ユーザ名またはパスワードを上書きする場合にのみ必要です（ステップ 3 を参照）。
ステップ 3	**iprcmdremote-username** ユーザ名 例： `Device(config)# ip rcmd remote-username Admin01`	（任意）リモート ユーザ名を指定します。
ステップ 4	**end** 例： `Device(config)# end`	（任意）コンフィギュレーション モードを終了します。この手順は、デフォルトのリモート ユーザ名またはパスワードを上書きする場合にのみ必要です（ステップ 3 を参照）。
ステップ 5	**copyrcp:**[[[//[*username@*]*location*]/*directory*] /*bundle_name*]**flash:** 例： `Device# copy rcp://netadmin@172.16.101.101/bundle1 flash:`	RCP を使用してネットワーク サーバからフラッシュ メモリ デバイスへコンフィギュレーション ファイルをコピーします。追加情報または確認を要求するデバイスからのプロンプトに対し応答します。このプロンプトは、**copy** コマンドで入力した情報量および **fileprompt** コマンドの現在の設定によって異なります。

TFTPサーバからフラッシュメモリデバイスへのコンフィギュレーション ファイルのコピー

TFTP サーバからフラッシュ メモリ デバイスへコンフィギュレーション ファイルをコピーするには、以下の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。

ステップ 2	**copytftp:**[[[*//location*]/*directory*]/bundle_name**flash:**
	例：
	`Device#`
	`copy`
	`tftp://at3k_cat9k_universalk9.SSA.03.12.02.EZP.150-12.02.EZP.150-12.02.EZP.bin`
	`flash:`
		TFTP サーバからフラッシュ メモリ デバイスへファイルをコピーします。追加情報または確認を要求するデバイスからのプロンプトに対し応答します。このプロンプトは、**copy** コマンドで入力した情報量および **file prompt** コマンドの現在の設定によって異なります。

例

次に、TFTP サーバから usbflash0 に挿入されているフラッシュ メモリ カードへ、switch-config という名前のコンフィギュレーション ファイルをコピーする例を示します。コピーされたファイルの名前は new-config に変更されます。

```
Device#
copy tftp:switch-config usbflash0:new-config
```

スタートアップ コンフィギュレーション ファイルでのコンフィギュレーション コマンドの再実行

スタートアップコンフィギュレーションファイルのコマンドを再実行するには、このセクションの手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configurememory** 例： `Device# configure memory`	スタートアップ コンフィギュレーション ファイルでコンフィギュレーション コマンドを再実行します。

スタートアップ コンフィギュレーションのクリア

スタートアップ コンフィギュレーションから設定情報を消去できます。デバイスをスタートアップ コンフィギュレーションなしで再起動した場合は、デバイスを最初から設定できるように、デバイスは、Setup コマンド ファシリティに移行します。スタートアップ コンフィギュレーションの内容をクリアするには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**erasenvram** 例：	スタートアップ コンフィギュレーションの内容をクリアします。

	コマンドまたはアクション	目的
	`Device# erase nvram`	（注） クラス A フラッシュ ファイル システムのプラットフォーム以外のすべてのプラットフォームでは、このコマンドにより NVRAM が消去されます。スタートアップ コンフィギュレーション ファイルは、いったん削除すると復元できません。クラス A フラッシュ ファイル システムのプラットフォーム上では、**erase startup-config** EXEC コマンドを使用すると、デバイスが CONFIG_FILE 環境変数により指定されたコンフィギュレーションを消去または削除します。この変数が NVRAM を指定している場合は、デバイスにより NVRAM が消去されます。CONFIG_FILE 環境変数がフラッシュ メモリ デバイスとコンフィギュレーション ファイル名を指定している場合は、デバイスによりコンフィギュレーション ファイルが削除されます。つまり、そのコンフィギュレーション ファイルは、デバイスにより消去されるのではなく、「削除済み」としてマークされます。この機能では、削除されたファイルを回復できます。

指定されたコンフィギュレーション ファイルの削除（CLI）

特定のフラッシュ デバイスの指定された設定を削除するには、このセクションの手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**delete** *flash-filesystem*:*filename* 例： `Device# delete usbflash0:myconfig`	特定のフラッシュ デバイス上の指定されたコンフィギュレーション ファイルを削除します。 （注） クラス A および B フラッシュ ファイル システムでは、フラッシュ メモリ内の特定のファイルを削除すると、そのファイルは削除済みとしてシステムによりマークされます。これにより、**undelete** EXEC コマンドを使用して、削除したファイルを後で回復できるようになります。消去されたファイルは回復できません。コンフィギュレーション ファイルを完全に消去するには、**squeeze** EXEC コマンドを使用します。クラス C フラッシュ ファイル システムでは、削除されたファイルは回復できません。CONFIG_FILE 環境変数で指定されたコンフィギュレーション ファイルを消去または削除しようとした場合、システムにより削除の確認を求めるプロンプトが表示されます。

クラス A フラッシュ ファイル システムでの CONFIG_FILE 環境変数の指定

クラス A フラッシュ ファイル システムでは、CONFIG_FILE 環境変数で指定されたスタートアップ コンフィギュレーション ファイルをロードするように Cisco IOS ソフトウェアを設定で

きます。CONFIG_FILE 変数のデフォルトは NVRAM になります。CONFIG_FILE 環境変数を変更するには、このセクションの手順を実行してください。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**copy**[*flash-url* | *ftp-url* | *rcp-url* | *tftp-url* | **system:running-config** | **nvram:startup-config**] *dest-flash-url* 例： `Device# copy system:running-config nvram:startup-config`	フラッシュ ファイル システムにコンフィギュレーション ファイルをコピーします。再起動時には、ここからデバイスにファイルがロードされます。
ステップ 3	**configureterminal** 例： `Device# configure terminal`	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 4	**bootconfig** *dest-flash-url* 例： `Device(config)# boot config 172.16.1.1`	CONFIG_FILE 環境変数を設定します。この手順により、実行時の CONFIG_FILE 環境変数が変更されます。
ステップ 5	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了します。
ステップ 6	**copysystem:running-confignvram:startup-config** 例： `Device# copy system:running-config nvram:startup-config`	スタートアップ コンフィギュレーションにステップ 3 で実行されたコンフィギュレーションを保存します。
ステップ 7	**showboot** 例： `Device# show boot`	（任意）CONFIG_FILE 環境変数の内容を確認できます。

例

次の例は、実行コンフィギュレーション ファイルをデバイスにコピーします。その後、システムが再起動されるとこの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として使用されます。

```
Device# copy system:running-config usbflash0:config2
Device# configure terminal
Device(config)# boot config usbflash0:config2
Device(config)# end
Device# copy system:running-config nvram:startup-config
[ok]
Device# show boot
BOOT variable = usbflash0:rsp-boot-m
CONFIG_FILE variable = nvram:
Current CONFIG_FILE variable = usbflash0:config2
Configuration register is 0x010F
```

次の作業

スタートアップ コンフィギュレーション ファイルの場所を指定すると、**nvram:startup-config** コマンドは、スタートアップ コンフィギュレーション ファイルの新しい場所のエイリアスとなります。**more nvram:startup-config** EXEC コマンドにより、スタートアップ コンフィギュレーションが、その場所に関係なく表示されます。**erase nvram:startup-config** EXEC コマンドにより、NVRAM の内容が消去され、CONFIG_FILE 環境変数で指定されたファイルが削除されます。

copy system:running-config nvram:startup-config コマンドを使用して設定を保存した場合、デバイスにより、コンフィギュレーション ファイルの完全バージョンはCONFIG_FILE環境変数で指定した場所に保存され、抽出バージョンはNVRAMに保存されます。抽出バージョンとは、アクセス リスト情報を含まないバージョンです。NVRAM に完全バージョンのコンフィギュレーション ファイルが含まれている場合、デバイスは、完全バージョンを抽出バージョンで上書きすることを確認するプロンプトを表示します。NVRAMに抽出コンフィギュレーションが含まれている場合、デバイスは確認のプロンプトを表示しないでNVRAMにある既存の抽出バージョンのコンフィギュレーション ファイルを上書きする処理を続行します。

（注） フラッシュ デバイスにあるファイルを CONFIG_FILE 環境変数として指定した場合、**copy system:running-config nvram:startup-config** コマンドでコンフィギュレーション ファイルを保存するたびに、古いコンフィギュレーション ファイルは「deleted」とマークされ、新しいコンフィギュレーション ファイルがそのデバイスに保存されます。それでも古いコンフィギュレーション ファイルがメモリを使用するため、最終的にフラッシュ メモリは一杯になります。**squeeze** EXEC コマンドを使用して、古いコンフィギュレーション ファイルを完全に削除してから、領域を再要求してください。

コンフィギュレーションファイルをダウンロードするデバイスの設定

ネットワーク コンフィギュレーションおよびホスト コンフィギュレーション ファイル名の順序付きリストを指定できます。Cisco IOS XE ソフトウェアは、適切なネットワークまたはホスト コンフィギュレーション ファイルをロードするまで、このリストをスキャンします。

システムの起動時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ようにデバイスを設定するには、次の項で説明する作業を少なくとも 1 つ実行します。

- ネットワーク コンフィギュレーション ファイルをダウンロードするデバイスの設定
- ホスト コンフィギュレーション ファイルをダウンロードするデバイスの設定

起動中にコンフィギュレーションファイルをロードできなかった場合、要求されたファイルがホストから提供されるまで、デバイスは 10 分ごと（デフォルト設定）に再試行します。試行が失敗するごとに、デバイスにより以下のメッセージがコンソール端末に表示されます。

```
Booting host-confg... [timed out]
```

スタートアップ コンフィギュレーション ファイルになんらかの問題がある場合、またはコンフィギュレーション レジスタが NVRAM を無視するように設定されている場合は、デバイスは Setup コマンド ファシリティに移行します。

ネットワーク コンフィギュレーション ファイルをダウンロードするデバイスの設定

起動時にサーバからネットワーク 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 Cisco IOS ソフトウェアを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	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 3	**bootnetwork** {**ftp:**[[[//[*username* [**:***password*]**@**]*location*]/*directory*]/*filename*] | **rcp:**[[[//[*username***@**]*location*]/*directory*]/*filename*] | **tftp:**[[[//*location*]/*directory*]/*filename*]} 例：	起動時にダウンロードするネットワーク コンフィギュレーション ファイルおよび使用されるプロトコル（TFTP、RCP、または FTP）を指定します。

	コマンドまたはアクション	目的
	`Device(config)# boot network tftp:hostfile1`	• ネットワーク コンフィギュレーション ファイル名を指定しない場合、Cisco IOS ソフトウェアはデフォルトのファイル名の network-confg を使用します。アドレスを省略した場合、デバイスはブロードキャスト アドレスを使用します。 • 複数のネットワーク コンフィギュレーション ファイルを指定できます。ソフトウェアは、ネットワーク コンフィギュレーション ファイルをロードできるまで、入力された順に試行します。この手順は、異なる設定情報を持つ、ネットワーク サーバ上にロードされるファイルを複数保持する場合に役立ちます。
ステップ 4	**serviceconfig** 例： `Device(config)# service config`	再起動時にネットワーク ファイルを自動的にロードするようにシステムをイネーブルにします。
ステップ 5	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了します。
ステップ 6	**copy system:running-config nvram:startup-config** 例： `Device# copy system:running-config nvram:startup-config`	実行コンフィギュレーションをスタートアップ コンフィギュレーション ファイルに保存します。

ホスト コンフィギュレーション ファイルをダウンロードするデバイスの設定

起動時にサーバからホスト 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 Cisco IOS ソフトウェアを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	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 3	**boothost**{**ftp:**[[[//[*username* [**:***password*]**@**]*location*]/*directory*]/*filename*] | **rcp:**[[[//[*username***@**]*location*]/*directory*]/*filename*] | **tftp:**[[[//*location*]/*directory*]/*filename*] } 例： `Device(config)# boot host tftp:hostfile1`	起動時にダウンロードするホスト コンフィギュレーション ファイルおよび使用されるプロトコル（FTP、RCP、または TFTP）を指定します。 • ホスト コンフィギュレーション ファイルの名前を指定しない場合、デバイスは、それ自身の名前を使用してホスト コンフィギュレーション ファイル名を形成します。このとき、その名前はすべて小文字に変換され、すべてのドメイン情報は削除され、「-config」が追加されます。ホスト名の情報を利用できない場合は、ソフトウェアはデフォルトのホスト コンフィギュレーション ファイル名の デバイス -confg を使用します。アドレスを省略した場合、デバイス はブロードキャスト アドレスを使用します。 • 複数のホストコンフィギュレーション ファイルを指定できます。Cisco IOS ソフトウェアは、ホスト コンフィギュレーションファイルをロードできるまで、入力された順に試行します。この手順は、異なる設定情報を持つ、ネットワーク サーバ上にロードされるファイルを複数保持する場合に役立ちます。

	コマンドまたはアクション	目的
ステップ 4	**serviceconfig** 例： `Device(config)# service config`	再起動時にホスト ファイルを自動的にロードするようにシステムをイネーブルにします。
ステップ 5	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了します。
ステップ 6	**copysystem:running-confignvram:startup-config** 例： `Device# copy system:running-config nvram:startup-config`	実行コンフィギュレーションをスタートアップ コンフィギュレーション ファイルに保存します。

例

次に、hostfile1 という名前のホスト コンフィギュレーション ファイルおよび networkfile1 という名前のネットワーク 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デバイスを設定する例を示します。デバイスは TFTP およびブロードキャスト アドレスを使用してファイルを取得します。

```
Device# configure terminal
Device(config)# boot host tftp:hostfile1
Device(config)# boot network tftp:networkfile1
Device(config)# service config
Device(config)# end
Device# copy system:running-config nvram:startup-config
```

その他の参考資料

関連資料

関連項目	参照先
Cisco IOS コマンド	『Cisco IOS Master Commands List, All Releases』
Cisco IOS コンフィギュレーション コマンド	『*Cisco IOS Configuration Fundamentals Command Reference*』

標準

標準	タイトル
この機能がサポートする新しい規格または変更された規格はありません。また、この機能による既存規格のサポートに変更はありません。	--

MIB

MIB	MIB リンク
• 新しい MIB または変更された MIB はサポートされていません。また、既存の MIB に対するサポートに変更はありません。	選択したプラットフォーム、Cisco ソフトウェア リリース、およびフィーチャ セットの MIB を検索してダウンロードする場合は、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

RFC

RFC	タイトル
新しい RFC または変更された RFC はサポートされていません。また、既存の RFC に対するサポートに変更はありません。	--

テクニカル サポート

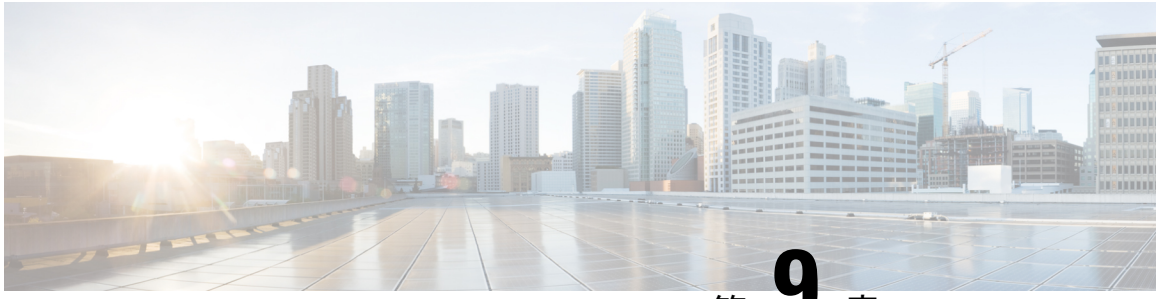
説明	リンク
右の URL にアクセスして、シスコのテクニカル サポートを最大限に活用してください。これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。この Web サイト上のツールにアクセスする際は、Cisco.com のログイン ID およびパスワードが必要です。	http://www.cisco.com/cisco/web/support/index.html

コンフィギュレーション ファイルの機能履歴と情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースの

みを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

リリース	変更箇所
Cisco IOS XE Everest 16.5.1a	この機能が導入されました。



第 9 章

コンフィギュレーションの置換とロールバック

コンフィギュレーションの置換とロールバックの前提条件

コンフィギュレーションの置換とロールバックの機能に対する入力となるコンフィギュレーション ファイルの形式は、標準の Cisco ソフトウェア コンフィギュレーション ファイルの、次に示すインデント規則に準拠している必要があります。

- 新しい行のすべてのコマンドは、コマンドがコンフィギュレーション サブモードにない限り、インデントなしで開始します。
- レベル 1 コンフィギュレーション サブモード内のコマンドは、スペース 1 個分インデントします。
- レベル 2 コンフィギュレーション サブモード内のコマンドは、スペース 2 個分インデントします。
- 以下、続くサブモード内のコマンドは、同じようにインデントします。

これらのインデント規則には、ソフトウェアが **show running-config** や **copy running-config** *destination-url* などのコマンドのコンフィギュレーション ファイルを作成する方法が記述されています。シスコ デバイスで生成されるコンフィギュレーション ファイルは、いずれもこうした規則に従います。

2つのコンフィギュレーションファイル（現在の実行コンフィギュレーションと、保存された置換用コンフィギュレーション）を合わせたサイズより大きな空きメモリが必要です。

コンフィギュレーションの置換とロールバックの制約事項

デバイスに、2 つのコンフィギュレーション ファイル（現在の実行コンフィギュレーションと、保存された置換用コンフィギュレーション）を合わせたサイズより大きな空きメモリがない場合、コンフィギュレーション置換操作は実行されません。

ネットワーク デバイスの物理コンポーネント（物理インターフェイスなど）に関連する特定の Cisco コンフィギュレーション コマンドは、実行コンフィギュレーションについて追加または削除することはできません。たとえば、コンフィギュレーション置換操作を行っても、そのインターフェイスがデバイス上に物理的に存在する場合、現在の実行コンフィギュレーションから **interface ethernet 0** コマンド行を削除することはできません。同様に、**interface ethernet 1** コマンド行は、そのようなインターフェイスがデバイス上に物理的に存在しない場合、実行コンフィギュレーションに追加することはできません。コンフィギュレーション置換操作でこのタイプの変更を試行すると、その特定のコマンド行が失敗したことを示すエラー メッセージが表示されます。

非常にまれなケースですが、ルータをリロードしないと特定の Cisco コンフィギュレーション コマンドを実行コンフィギュレーションから削除できないことがあります。コンフィギュレーション置換操作でこのタイプのコマンドの削除を試行すると、その特定のコマンド行が失敗したことを示すエラー メッセージが表示されます。

コンフィギュレーションの置換とロールバックについて

設定アーカイブ（Configuration Archive）

Cisco IOS コンフィギュレーション アーカイブは、**configure replace** コマンドを使用するコンフィギュレーションのロールバック機能を強化するために、Cisco IOS コンフィギュレーション ファイルのアーカイブの保存、編成、管理を行うことを目的とした機能です。この機能の導入前にも、実行コンフィギュレーションのコピーを **copy running-config** *destination-url* コマンドを使用して保存し、ローカルやリモートに置換ファイルを保管できました。ただし、この方法ではファイルの自動管理を行うことはできませんでした。一方、コンフィギュレーションの置換とロールバック機能では、実行コンフィギュレーション ファイルを自動的に Cisco IOS コンフィギュレーション アーカイブに保存できます。アーカイブされたファイルはコンフィギュレーションのチェックポイントとして参照することができ、**configure replace** コマンドを使用して以前のコンフィギュレーション状態に戻すために利用できます。

archive config コマンドを使用すると、Cisco IOS コンフィギュレーションをコンフィギュレーション アーカイブに保存できます。その場合、標準のディレクトリとファイル名のプレフィク

スが使用され、バージョン番号（およびオプションでタイムスタンプ）が自動的に付加されます。バージョン番号は連続したファイルを保存するごとに、1 つずつ大きくなります。この機能により、保存した Cisco IOS コンフィギュレーション ファイルを一貫して識別できます。アーカイブに保存する実行コンフィギュレーションの数は指定することができます。アーカイブ内のファイル数が上限値に達すると、次に最新のファイルが保存されるときに、最も古いファイルが自動的に消去されます。**show archive** コマンドを使用すると、Cisco IOS コンフィギュレーション アーカイブに保存されているすべてのコンフィギュレーション ファイルに関する情報が表示されます。

コンフィギュレーション ファイルを保存する Cisco IOS コンフィギュレーション アーカイブは、**configure replace** コマンドで使用することによって、FTP、HTTP、RCP、TFTP のファイル システム上に配置できます。

コンフィギュレーションの置換

configurereplace 特権 EXEC コマンドにより、現在の実行コンフィギュレーションを、保存しておいた Cisco IOS コンフィギュレーション ファイルで置換することができます。この機能は、コンフィギュレーションを保存しておいた状態へ戻すために使用することができ、そのコンフィギュレーション状態が保存された後にどのような変更が加えられても、効果的にロールバックさせることができます。

configurereplace コマンドを使用するときは、現在の実行コンフィギュレーションと置換するための、保存された Cisco IOS コンフィギュレーション ファイルを指定する必要があります。置換ファイルは、Cisco IOS デバイスによって作成された完全なコンフィギュレーション（**copyrunning-config** *destination-url* コマンドによって作成されたものなど）であることが必要です。あるいは、置換ファイルを外部的に作成する場合は Cisco IOS デバイスが作成するファイル形式に完全に準拠していなければなりません。**configurerepl**ace コマンドを入力すると、現在の実行コンフィギュレーションが指定された置換コンフィギュレーションと比較され、一連の diff が生成されます。2 つのファイルの比較に使用されるアルゴリズムは、**showarchiveconfigdifferences** コマンドで使用されるものと同じです。置換コンフィギュレーションの状態になるよう、diff の結果が Cisco IOS パーサーによって適用されます。diff のみが適用されるため、現在の実行コンフィギュレーション上にすでに存在していた設定コマンドを再適用することにより生じる、潜在的なサービスの中断を避けられます。このアルゴリズムでは、順序に依存するコマンド（アクセス リストなど）へのコンフィギュレーション変更を、複数のパスプロセスを通して効果的に実行します。通常の環境では、コンフィギュレーション置換操作の完了に必要なパスは 3 つまでであり、ループ動作を防ぐためのパスは最大 5 つまでに制限されます。

Cisco IOS **copy** *source-url***running-config** 特権 EXEC コマンドは、保存された Cisco IOS コンフィギュレーション ファイルを実行コンフィギュレーションへコピーするためによく使用されます。**copy** *source-url***running-config** コマンドを **configurerepl**ace *target-url* 特権 EXEC コマンドの代わりに使用する場合は、次の大きな違いに注意が必要です。

- **copy** *source-url***running-config** コマンドはマージ動作であり、ソース ファイルと現在の実行コンフィギュレーションの両方のコマンドがすべて保持されます。このコマンドでは、現在の実行コンフィギュレーションにのみ含まれ、ソースファイルには存在しないコマンドが削除されることはありません。これに対し、**configurerepl**ace *target-url* コマンドでは、

置換ファイルに存在しないコマンドが現在の実行コンフィギュレーションから削除され、追加する必要のあるコマンドが現在の実行コンフィギュレーションに追加されます。

- **copy** *source-url***running-config** コマンドでは、現在の実行コンフィギュレーションにすでに存在しているかどうかにかかわらず、ソースファイル中のすべてのコマンドが適用されます。このアルゴリズムは効率的でない上、場合によってはサービスの停止が発生します。これに対し、**configure replace** *target-url* コマンドでは適用が必要なコマンドのみを適用し、現在の実行コンフィギュレーションに存在しているコマンドは再適用されません。
- **copy** *source-url* **running-config** コマンドでは部分的なコンフィギュレーション ファイルもコピー元として使用できますが、**configure replace** *target-url* コマンドの置換ファイルとして使用できるのは、完全な Cisco IOS コンフィギュレーション ファイルのみです。

コンフィギュレーション置換操作にロック機能が導入されました。**configure replace** コマンドが使用されると、コンフィギュレーション置換中、デフォルトで実行コンフィギュレーションファイルがロックされます。このロック メカニズムによって、置換動作の実行中に他のユーザが実行コンフィギュレーションを変更しようとしたために、置換動作の不正終了が発生することを防止できます。**no lock** キーワードを **configure replace** コマンドの実行時に使用すると、実行コンフィギュレーションのロックをディセーブルにできます。

実行コンフィギュレーションのロックは、コンフィギュレーションの置換動作終了時に自動的にクリアされます。**show configuration lock** コマンドを使用すると、現在実行コンフィギュレーションに適用されているロックをすべて表示できます。

設定のロール バック

ロールバックの概念は、データベースの操作ではトランザクション プロセス モデルに由来します。データベース トランザクションでは、あるデータベースのテーブルに一連の変更を加えることがあります。その後、変更を実行する（変更を恒久的に適用する）か、変更をロールバックする（変更を破棄してテーブルを以前の状態に戻す）かを選択することになります。ここでロールバックが意味するのは、変更のログを含んだジャーナルファイルが破棄され、何の変更も加えられないということです。ロールバック操作の結果として、加えた変更が適用される前の状態に戻ります。

configurereplace コマンドを使用することで、以前のコンフィギュレーション状態へ戻ることが可能になり、コンフィギュレーション状態の保存後に加えた変更を効率的にロールバックさせることができます。Cisco IOS コンフィギュレーション ロールバックは、適用された一連の変更をもとにロールバック動作を行うのではなく、保存された Cisco コンフィギュレーションファイルに基づいた特定のコンフィギュレーション状態へ戻るというコンセプトを採用しています。このコンセプトは、チェックポイント（データベースの保存されたバージョン）に特定の状態を保存しておくという、データベースの考え方に類似しています。

コンフィギュレーションのロールバック機能が必要な場合、コンフィギュレーションの変更に先立って Cisco IOS 実行コンフィギュレーションを保存する必要があります。次に、コンフィギュレーションを変更した後に（**configurereplace** *target-url* コマンドを使用し）保存したコンフィギュレーション ファイルを使って変更をロール バックします。保存された Cisco IOS コン

フィギュレーション ファイルならどれでも置換コンフィギュレーションとして指定できるため、一部のロールバック モデルのように、ロールバックの数が制限されることもありません。

コンフィギュレーション ロールバック変更確認

コンフィギュレーション ロールバック変更確認機能により、コンフィギュレーション変更の実行に際して確認を要求するようオプションで設定できます。この確認が受信できない場合、コンフィギュレーションは変更が適用される前の状態に戻されます。このメカニズムは、ネットワーク デバイスとユーザまたは管理アプリケーションとの接続において、コンフィギュレーション変更に起因する切断を防止するものです。

コンフィギュレーションの置換とロールバックの利点

- コンフィギュレーションの変更を効率的にロールバックさせて、以前のコンフィギュレーション状態へ戻ることが可能。
- デバイスをリロードしたり、CLI で実行コンフィギュレーション ファイルに加えた変更を手動で元に戻したりすることなく、現在の実行コンフィギュレーション ファイルをスタートアップ コンフィギュレーション ファイルと置換できるため、システムのダウンタイムが減少。
- 保存しておいたどの Cisco IOS コンフィギュレーション状態に戻すことも可能。
- 追加や削除が必要なコマンドだけが影響を受ける場合、デバイスに完全なコンフィギュレーション ファイルを適用することができるため、コンフィギュレーションの変更が簡素化。
- **configurerepl**ace コマンドを **copy** *source-url***running-config** コマンドの代用として使用する場合、現在の実行コンフィギュレーションに存在しているコマンドを再度適用することがないため、効率が向上し、かつサービス停止のリスクを回避。

コンフィギュレーションの置換とロールバックの使用方法

コンフィギュレーション アーカイブの作成（CLI）

configure replace コマンドを使用するために、前提条件となる設定はありません。**configure replace** コマンドと Cisco IOS コンフィギュレーション アーカイブおよび **archive config** コマンドとの併用は任意ですが、コンフィギュレーション ロールバックの使用にあたっては大きな利点があります。**archive config** コマンドを使用する前に、コンフィギュレーション アーカイブを設定しておく必要があります。コンフィギュレーション アーカイブの特性を設定するには、次の作業を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	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 3	**archive** 例： `Device(config)# archive`	アーカイブ コンフィギュレーション モードを開始します。
ステップ 4	**path** *URL* 例： `Device(config-archive)# path flash:myconfiguration`	Cisco IOS コンフィギュレーション アーカイブの場所と、ファイル名のプレフィックスを指定します。 （注） パスのところでファイルの代わりにディレクトリを指定する場合、ディレクトリ名は path flash:/directory/ のように後ろにスラッシュを付ける必要があります。このスラッシュはファイル名の後ろでは必要ありません。ディレクトリを指定する場合にだけ使います。
ステップ 5	**maximum** *number* 例： `Device(config-archive)# maximum 14`	（任意）Cisco IOS コンフィギュレーション アーカイブに保存される実行コンフィギュレーションのアーカイブ ファイル数の上限値を設定します。 • *number* 引数は、Cisco IOS コンフィギュレーション アーカイブに保存される実行コンフィギュレーションのアーカイブ ファイル数の上限値を示します。有効な値は 1 ～ 14 です。デフォルトは 10 です。

	コマンドまたはアクション	目的
		（注） このコマンドを使用する前に、**path** コマンドを設定して Cisco IOS コンフィギュレーション アーカイブの位置とファイル名プレフィックスを指定しておく必要があります。
ステップ 6	**time-period** 分 例： `Device(config-archive)# time-period 1440`	（任意）Cisco IOS コンフィギュレーション アーカイブに実行コンフィギュレーションのアーカイブ ファイルを自動保存する間隔を設定します。 • Cisco IOS コンフィギュレーション アーカイブに現在の実行コンフィギュレーションのアーカイブ ファイルをどれほどの頻度で自動保存するかを、minutes 引数により分単位で指定します。 （注） このコマンドを使用する前に、**path** コマンドを設定して Cisco IOS コンフィギュレーション アーカイブの位置とファイル名プレフィックスを指定しておく必要があります。
ステップ 7	**end** 例： `Device(config-archive)# end`	特権 EXEC モードに戻ります。
ステップ 8	**archiveconfig** 例： `Device# archive config`	現在の実行コンフィギュレーションファイルをコンフィギュレーション アーカイブに保存します。 （注） このコマンドを使用する前に、**path** コマンドを設定する必要があります。

コンフィギュレーションの置換またはロールバックの実行（CLI）

保存された Cisco IOS コンフィギュレーション ファイルで現在の実行コンフィギュレーション ファイルを置換するには、次の作業を実行します。

（注） この手順の前に、コンフィギュレーションアーカイブを作成しておく必要があります。詳細については、コンフィギュレーション アーカイブの作成（CLI）を参照してください。次に、現在の実行コンフィギュレーションで問題が生じた場合に、アーカイブしておいたコンフィギュレーションに戻す手順の詳細を示します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configurereplace** *target-url* [**nolock**] [**list**] [**force**] [**ignore case**] [**revert trigger** ［**error** ］［**timer** *minutes*］｜**time** *minutes*]] 例： `Device# configure replace flash: startup-config time 120`	保存しておいた Cisco IOS コンフィギュレーション ファイルで現在の実行コンフィギュレーション ファイルを置換します。 • *target - url* 引数は、**archive config** コマンドで作成されたコンフィギュレーション ファイルなど、現在の実行コンフィギュレーションと置換する、保存された Cisco IOS コンフィギュレーション ファイルの URL です（Cisco IOS ファイル システムでアクセス可能なもの）。 • **list** キーワードは、コンフィギュレーション置換動作のパスごとに、Cisco IOS ソフトウェア パーサーによって適用されるコマンド ラインのリストを表示します。実行されたパスの総数も表示されます。 • **force** キーワードは、現在の実行コンフィギュレーションから指定した Cisco IOS コンフィギュレーション ファイルへの置換を、確認プロンプトを出さずに実行します。

	コマンドまたはアクション	目的

		• **time***minutes* キーワードおよび引数は、現在の実行コンフィギュレーション ファイルの置換確認のために **configure confirm** コマンドを入力する制限時間（分単位）を指定します。**configure confirm** コマンドが指定の制限時間内に入力されない場合、コンフィギュレーション置換操作は自動的に戻されます（つまり、現在の実行コンフィギュレーションファイルが **configure replace** コマンド入力以前のコンフィギュレーション状態へと回復されます）。
		• **nolock** キーワードは、コンフィギュレーション置換操作中に他のユーザが実行コンフィギュレーションを変更しないように実行コンフィギュレーション ファイルをロックする機能をオフにします。
		• **revert trigger** キーワードは、元のコンフィギュレーションへ戻すトリガーを次の内容から設定します。
		  • **error**：エラー時に元のコンフィギュレーションに戻します。
		  • **timer***minutes*：指定した時間が過ぎると元のコンフィギュレーションに戻します。
		• **ignore case** キーワードで、コンフィギュレーションに確認コマンドの大文字と小文字の区別を無視させることができます。

ステップ 3	**configurerevert** {**now**|**timer** {*minutes*| **idle** *minutes*} } 例： `Device# configure revert now`	（任意）時間指定ロールバックをキャンセルしてロールバックを即時トリガーする、または時間指定ロールバックのパラメータをリセットするには、特権EXECモードで **configure revert** コマンドを使用します。

	コマンドまたはアクション	目的

		• **now**：ロールバックをただちにトリガーします。
		• **timer**：コンフィギュレーションを元に戻すタイマーをリセットします。
		  • 元に戻す時間を分単位で新たに指定するには、*minutes* 引数を **timer** キーワードとともに使用します。
		  • 保存されたコンフィギュレーションに戻すまでに、操作が行われないアイドル時間を最大どれほど長く許容できるかを設定するには、分単位の時間とともに **idle** キーワードを使用します。

ステップ 4	**configureconfirm** 例： `Device# configure confirm`	（任意）保存しておいた Cisco IOS コンフィギュレーション ファイルの現在の実行コンフィギュレーション ファイルへの置換を確認します。 （注） このコマンドは、**time** *seconds* キーワードと **configure replace** コマンドの引数が指定されているときのみ使用します。
ステップ 5	**exit** 例： `Device# exit`	ユーザ EXEC モードに戻ります。

機能のモニタリングおよびトラブルシューティング（CLI）

コンフィギュレーションの置換とロールバック機能をモニタおよびトラブルシューティングするには、この手順を実行します。

手順

ステップ 1 **enable**

このコマンドを使用して、特権EXECモードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。

例：

```
Device> enable
Device#
```

ステップ 2 showarchive

Cisco IOS コンフィギュレーション アーカイブに保存されているファイルに関する情報を表示するには、次のコマンドを使用します。

例：

```
Device# show archive
There are currently 1 archive configurations saved.
The next archive file will be named flash:myconfiguration-2
 Archive #  Name
   0
   1       flash:myconfiguration-1 <- Most Recent
   2
   3
   4
   5
   6
   7
   8
   9
   10
   11
   12
   13
   14
```

次に、実行コンフィギュレーションのアーカイブファイルをいくつか保存した状態で**showarchive**コマンドを使用した場合の出力例を示します。この例では、保存されるアーカイブファイルの最大数が 3 に設定されています。

例：

```
Device# show archive
There are currently 3 archive configurations saved.
The next archive file will be named flash:myconfiguration-8
 Archive #  Name
   0
   1       :Deleted
   2       :Deleted
   3       :Deleted
   4       :Deleted
   5       flash:myconfiguration-5
   6       flash:myconfiguration-6
   7       flash:myconfiguration-7 <- Most Recent
   8
   9
   10
   11
   12
   13
   14
```

ステップ 3 debugarchiveversioning

このコマンドを使用して、Cisco IOS コンフィギュレーション アーカイブのアクティビティのデバッグを有効にして、コンフィギュレーションの置換とロールバックをモニタおよびトラブルシューティングします。

例：

```
Device# debug archive versioning
Jan  9 06:46:28.419:backup_running_config
Jan  9 06:46:28.419:Current = 7
Jan  9 06:46:28.443:Writing backup file flash:myconfiguration-7
Jan  9 06:46:29.547: backup worked
```

ステップ 4 debugarchiveconfigtimestamp

このコマンドを使用して、コンフィギュレーション置換操作の各必須段階の処理時間、および操作中のコンフィギュレーション ファイルのサイズのデバッグをイネーブルにします。

例：

```
Device# debug archive config timestamp
Device# configure replace flash:myconfiguration force
Timing Debug Statistics for IOS Config Replace operation:
       Time to read file usbflash0:sample_2.cfg = 0 msec (0 sec)
       Number of lines read:55
       Size of file        :1054
Starting Pass 1
       Time to read file system:running-config = 0 msec (0 sec)
       Number of lines read:93
       Size of file        :2539
       Time taken for positive rollback pass = 320 msec (0 sec)
       Time taken for negative rollback pass = 0 msec (0 sec)
       Time taken for negative incremental diffs pass = 59 msec (0 sec)
       Time taken by PI to apply changes = 0 msec (0 sec)
       Time taken for Pass 1 = 380 msec (0 sec)
Starting Pass 2
       Time to read file system:running-config = 0 msec (0 sec)
       Number of lines read:55
       Size of file        :1054
       Time taken for positive rollback pass = 0 msec (0 sec)
       Time taken for negative rollback pass = 0 msec (0 sec)
       Time taken for Pass 2 = 0 msec (0 sec)
Total number of passes:1
Rollback Done
```

ステップ 5 exit

このコマンドを使用して、ユーザ EXEC モードに戻ります。

例：

```
Device# exit
Device>
```

コンフィギュレーションの置換とロールバックの設定例

コンフィギュレーション アーカイブの作成

次の例は、Cisco IOS コンフィギュレーション アーカイブの初期設定を実行する方法を示しています。この例では、flash:myconfiguration がコンフィギュレーション アーカイブの保存位置およびファイル名のプレフィックスとして設定され、保存するアーカイブ ファイルが最大 10 個に設定されます。

```
configure terminal
!
archive
 path flash:myconfiguration
 maximum 10
end
```

現在の実行コンフィギュレーションを保存された Cisco IOS コンフィギュレーション ファイルで置換

次の例では、flash:myconfiguration という名前で保存された Cisco IOS コンフィギュレーション ファイルで現在の実行コンフィギュレーションを置換する方法を示します。**configurereplace** コマンドでは、確認プロンプトでインタラクティブに操作を進めます。

```
Device# configure replace flash:myconfiguration
This will apply all necessary additions and deletions
to replace the current running configuration with the
contents of the specified configuration file, which is
assumed to be a complete configuration, not a partial
configuration. Enter Y if you are sure you want to proceed. ? [no]: Y
Total number of passes: 1
Rollback Done
```

次の例では、コンフィギュレーション置換操作中に適用されるコマンドラインを表示するために、**list** キーワードを指定しています。

```
Device# configure replace flash:myconfiguration list
This will apply all necessary additions and deletions
to replace the current running configuration with the
contents of the specified configuration file, which is
assumed to be a complete configuration, not a partial
configuration. Enter Y if you are sure you want to proceed. ? [no]: Y
!Pass 1
!List of Commands:
no snmp-server community public ro
snmp-server community mystring ro

end
Total number of passes: 1
Rollback Done
```

スタートアップ コンフィギュレーション ファイルへの復帰

次の例に、**configure replace** コマンドを使用して Cisco IOS スタートアップ コンフィギュレーション ファイルへ復帰する方法を示します。この例は、オプションの **force** キーワードを使用して、インタラクティブ ユーザ プロンプトをオーバーライドする方法を示しています:

```
Device# configure replace flash:startup-config force
Total number of passes: 1
Rollback Done
```

configure confirm コマンドを使用したコンフィギュレーション置換操作の実行

次に、**configure replace** コマンドを **time** *minutes* キーワードおよび引数と共に使用する例を示します。現在実行中のコンフィギュレーション ファイルの置換を実行するには、指定の制限時間内に **configure confirm** コマンドを入力する必要があります。**configure confirm** コマンドが指定の制限時間内に入力されない場合、コンフィギュレーション置換操作は自動的に戻されます（つまり、現在実行中のコンフィギュレーション ファイルが **configure replace** コマンド入力以前のコンフィギュレーション状態へと回復されます）。

```
Device# configure replace flash:startup-config time 120
This will apply all necessary additions and deletions
to replace the current running configuration with the
contents of the specified configuration file, which is
assumed to be a complete configuration, not a partial
configuration. Enter Y if you are sure you want to proceed. ? [no]: Y
Total number of passes: 1
Rollback Done
Device# configure confirm
```

次に、**configure revert** コマンドを **timer** キーワードとともに使用する例を示します。時間指定ロールバックをキャンセルしてロールバックを即時トリガーする、または時間指定ロールバックのパラメータをリセットするには、**configure revert** コマンドを入力する必要があります。

```
Device# configure revert timer 100
```

コンフィギュレーション ロールバック操作の実行

次の例は、現在実行中のコンフィギュレーションへの変更を行い、その変更をロールバックする方法を示しています。コンフィギュレーション ロールバック操作の一部として、ファイルに変更を加える前に現在の実行コンフィギュレーションを保存する必要があります。この例では、現在実行中のコンフィギュレーションの保存に **archive config** コマンドが使用されています。**configure replace** コマンドで生成された出力は、ロールバック操作を完了するために 1 つのパスのみが実行されたことを示します。

（注） **archive config** コマンドを使用する前に、**path** コマンドで Cisco IOS コンフィギュレーション アーカイブのファイルの位置とファイル名のプレフィックスを指定する必要があります。

次のように、設定アーカイブの現在実行中のコンフィギュレーションを保存します。

```
archive config
```

それから、次の例に示すようにコンフィギュレーションの変更を入力します。

```
configure terminal
!
user netops2 password rain
user netops3 password snow
exit
```

実行コンフィギュレーション ファイルに変更を加えた後、それらの変更をロールバックさせて、変更前のコンフィギュレーションに戻したくなったとします。**show archive** コマンドは、交換ファイルとして使用される設定のバージョンを確認するために使用されます。次の例に示すように、**configure replace** コマンドは交換コンフィギュレーション ファイルへ戻すために使用されます。

```
Device# show archive
There are currently 1 archive configurations saved.
The next archive file will be named flash:myconfiguration-2
 Archive #  Name
   0
   1       flash:myconfiguration-1 <- Most Recent
   2
   3
   4
   5
   6
   7
   8
   9
   10
Device# configure replace flash:myconfiguration-1
Total number of passes: 1
Rollback Done
```

コンフィギュレーションの置換およびコンフィギュレーションのロールバックの機能履歴と情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースのみを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

リリース	変更箇所
Cisco IOS XE Everest 16.5.1a	この機能が導入されました。

第 10 章

ソフトウェア メンテナンス アップグレード

ソフトウェア メンテナンス アップグレード（SMU）は、システムにインストールしてパッチ修正やセキュリティ解決をリリースされたイメージに提供できるパッケージです。

ソフトウェア メンテナンス アップグレードの制約事項

In Service Software Upgrade（ISSU）はサポートされません。

ソフトウェア メンテナンス アップグレードについて

SMU の概要

ソフトウェア メンテナンス アップグレード（SMU）は、システムにインストールしてパッチ修正やセキュリティ解決をリリースされたイメージに提供できるパッケージです。

SMU パッケージはリリースごとおよびコンポーネントごとに提供され、プラットフォームに固有です。

SMU はネットワークの問題に迅速に対応できるようにするとともに、必要なテストの時間と範囲を削減するため、従来の IOS ソフトウェアには多大なメリットがあります。Cisco IOS XE プラットフォームでは SMU の互換性を内部的に検証し、互換性のない SMU はインストールできません。

すべて SMU が後続の Cisco IOS XE ソフトウェア メンテナンス リリースに統合されています。SMU は独立した自己完結型パッケージであり、前提条件や依存関係はありません。SMU はどのような順序でもインストールまたはアンインストールできます。

Cisco IOS XE Everest 16.6.1 以降、SMU は拡張メンテナンス リリースでのみ、基盤となるソフトウェア リリースのライフサイクルにわたってサポートされます。

次に、SMU をインストールする 3 つの基本ステップを示します。

- ファイルシステムへの SMU の追加。
- システムでの SMU のアクティブ化。
- リロードが繰り返されても持続させるための SMU の変更のコミット。

ソフトウェア メンテナンス アップ グレード パッケージ

SMU パッケージには、パッケージの内容を記述するいくつかのメタデータとともに、リリースにパッチを適用するための一連のファイルがいくつか含まれています。

SMU のワークフロー

SMU プロセスは、SMU Committee への要求によって開始されます。カスタマー サポートに連絡し、SMU 要求を行います。

SMU パッケージがリリースされると [シスコ ソフトウェアのダウンロード（Cisco Software Download）]ページに掲載されます。そのパッケージをダウンロードし、インストールします。

SMU パッケージ

SMU パッケージには、SMU が要求されている報告済みの問題のメタデータと修正が含まれています。

SMU のリロード

SMUのタイプは、SMUのインストール後のシステムへの影響を説明します。SMUはトラフィックに影響を与えない場合もありますが、デバイスの再起動、リロード、スイッチオーバーを引き起こす可能性もあります。

すべての SMU で、アクティブ化中にシステムをコールド リロードする必要があります。コールド リロードは、オペレーティング システムを完全にリロードします。このアクションは、リロードの間（現在は最大 5 分間）、トラフィック フローに影響します。このリロードにより、SMU の一部としてインストールされている正しいライブラリとファイルですべてのプロセスが起動します。

ソフトウェア メンテナンスの更新の管理方法

SMU パッケージの管理

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**install add file bootflash:** *filename* 例： `Device# install add file bootflash:isr4300-universalk9.BLD_POLARIS_DEV_SMU_LATEST_20170128_040557.1.CSCvbXXXXX.SSA.smu.bin`	メンテナンス更新プログラム パッケージをリモート ロケーションからデバイスにコピーし、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。 • このコマンドは、ファイルで基本的な互換性チェックを実行し、SMU パッケージがプラットフォームでサポートされていることを確認します。また、パッケージ/SMU.sta ファイル内にエントリを追加することで、ステータスを監視し、維持できるようにします。
ステップ 3	**install activate file bootflash:** *filename* 例： `Device# install activate file bootflash:isr4300-universalk9.BLD_POLARIS_DEV_SMU_LATEST_20170128_040557.1.CSCvbXXXXX.SSA.smu.bin`	互換性チェックを実行し、パッケージをインストールして、パッケージのステータスの詳細を更新します。 • 再起動可能なパッケージの場合、このコマンドは適切なポストインストール スクリプトをトリガーして必要なプロセスを再起動します。また、再起動できないパッケージの場合は、リロードをトリガーします。

	コマンドまたはアクション	目的
ステップ 4	**install commit file bootflash:** *filename* 例： `Device# install commit file bootflash:isr4300-universalk9.BLD_POLARIS_DEV_SMU_LATEST_20170128_040557.1.CSCvbXXXXX.SSA.smu.bin`	リロードが繰り返されても持続するようにアクティブ化の変更をコミットします。 • アクティブ化の後で、システムがアップしている間、または最初のリロード後にコミットできます。パッケージがアクティブになっていてもコミットされていない場合は、最初のリロード後はアクティブの状態を保ちますが、2回目のリロード後はアクティブ状態を保ちません。
ステップ 5	**install rollback to** {**base** | **committed** | **id** *commit-ID*} 例： `Device# install rollback to committed`	デバイスを以前のインストール状態に戻します。 • ロールバック後にリロードする必要があります。
ステップ 6	**install deactivate file bootflash:** *filename* 例： `Device# install deactivate file bootflash:isr4300-universalk9.BLD_POLARIS_DEV_SMU_LATEST_20170128_040557.1.CSCvbXXXXX.SSA.smu.bin`	アクティブなパッケージを非アクティブ化し、パッケージステータスを更新し、再起動またはリロードするプロセスをトリガーします。
ステップ 7	**install remove** {**file bootflash:** *filename* | **inactive**} 例： `Device# install remove file bootflash:isr4300-universalk9.BLD_POLARIS_DEV_SMU_LATEST_20170128_040557.1.CSCvbXXXXX.SSA.smu.bin`	すべての、または指定した非アクティブなSMUパッケージをファイル システムから削除します。
ステップ 8	**show version** 例： `Device# show version`	デバイスのイメージ バージョンを表示します。
ステップ 9	**show install summary** 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。 • このコマンドの出力は、設定されている **install** コマンドに応じて変化します。

ソフトウェア メンテナンス アップグレードの設定例

例：SMU の管理

次に、SMU ファイルをブートフラッシュにコピーする例を示します。

```
Device# copy tftp://172.19.1.250//auto/tftpboot/user/isr4300-
universalk9.BLD_POLARIS_DEV_SMU_LATEST_20170128_040557.1.CSCvbXXXXX.SSA.smu.bin

bootflash:
Destination filename [isr4300-
universalk9.BLD_POLARIS_DEV_SMU_LATEST_20170128_040557.1.CSCvbXXXXX.SSA.smu.
bin]?

Accessing tftp://172.19.1.250//auto/tftpboot/folder1/isr4300-
universalk9.BLD_POLARIS_DEV_SMU_LATEST_20170128_040557.1.CSCvbXXXXX.SSA.smu.bin...
Loading /auto/tftpboot/folder1/isr4300-
universalk9.BLD_POLARIS_DEV_SMU_LATEST_20170128_040557.1.CSCvbXXXXX.SSA.smu.bin from
172.19.1.250 (via GigabitEthernet0): !
[OK - 17668 bytes]
17668 bytes copied in 0.058 secs (304621 bytes/sec)
```

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

```
Device# show install summary

Active Packages:
No packages
Inactive Packages:
No packages
Committed Packages:
No packages
Uncommitted Packages:
No packages
```

次に、メンテナンス更新プログラム パッケージ ファイルを追加する例を示します。

```
Device# install add file bootflash:isr4300-
universalk9.BLD_POLARIS_DEV_SMU_LATEST_20170128_040557.1.CSCvbXXXXX.SSA.smu.bin

install_add: START Sat Feb 26 14:06:04 PST 2017
SUCCESS: install_add /bootflash/isr4300-
universalk9.BLD_POLARIS_DEV_SMU_LATEST_20170128_040557.1.CSCvbXXXXX.SSA.smu.bin Sat Feb
26 14:06:12 PST 2017
Device#
```

次に、SMU パッケージ ファイルをデバイスに追加した後の **showinstallsummary** コマンドの出力例を示します。

```
Device# show install summary

Active Packages:
No packages
Inactive Packages:
bootflash: isr4300-universalk9.BLD_POLARIS_DEV_SMU_LATEST_20170128_040557.1.
```

```
CSCvbXXXXX.SSA.smu.bin
Committed Packages:
No packages
Uncommitted Packages:
No packages
Device#
```

次に、追加した SMU 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 file bootflash:
isr4300-universalk9.BLD_POLARIS_DEV_SMU_LATEST_20170128_040557.1.
CSCvbXXXXX.SSA.smu.bin

install_activate: START Sat Feb 26 14:10:55 PST 2017
The activation step would require a reload. Do you want to proceed? [y/n]y
Regular SMU. Reloading the box to complete activation of the SMU...
Feb 26 14:11:23.873 R0/0: %PMAN-5-EXITACTION: Process manager is exiting:
reload action requested
Initializing Hardware ...
Checking for PCIe device presence...done
System integrity status: 0x610
Rom image verified correctly
<after reload>
Device#
```

次に、**show version** コマンドの出力例を示します。

```
Device# show version

Cisco IOS XE Software, Version BLD_POLARIS_DEV_SMU_LATEST_20170127_200213 -
SMU-PATCHED
Cisco IOS Software [Everest], ISR Software (X86_64_LINUX_IOSD-UNIVERSALK9-M),
Experimental Version 16.6.20170127:201839 [polaris_dev-BLDBLD_
POLARIS_DEV_SMU_LATEST_20170127_200213 110]
Copyright (c) 1986-2017 by Cisco Systems, Inc.
Compiled Sat 26-Feb-17 16:07 by mcpre
...
```

次に示すのは、**showinstallsummary** コマンドがモデル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

Active Packages:
bootflash:isr4300-universalk9.BLD_POLARIS_DEV_SMU_LATEST_20170128_040557.1.
CSCvbXXXXX.SSA.smu.bin
Inactive Packages:
No packages
Committed Packages:
No packages
Uncommitted Packages:
bootflash:isr4300-universalk9.BLD_POLARIS_DEV_SMU_LATEST_20170128_040557.1.
CSCvbXXXXX.SSA.smu.bin
Device#
```

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

```
Device# show install active

Active Packages:
bootflash:isr4300-universalk9.BLD_POLARIS_DEV_SMU_LATEST_20170128_040557.1.
```

```
CSCvbXXXXX.SSA.smu.bin
```

次の例では、**installcommit** コマンドの実行方法を示します。

```
Device# install commit

install_commit: START Sat Feb 26 06:46:48 UTC 2017
SUCCESS: install_commit Sat Feb 26 06:46:52 UTC 2017
Device#
```

次に示すのは、**showinstallsummary** コマンドが、更新プログラム パッケージがコミットされてリロードが繰り返されても持続することを表示する場合の出力例です。

```
Device# show install summary

Active Packages:
bootflash:isr4300-universalk9.BLD_POLARIS_DEV_SMU_LATEST_20170128_040557.1.
CSCvbXXXXX.SSA.smu.bin
Inactive Packages:
No packages
Committed Packages:
bootflash:isr4300-universalk9.BLD_POLARIS_DEV_SMU_LATEST_20170128_040557.1.
CSCvbXXXXX.SSA.smu.bin
Uncommitted Packages:
No packages
Device#
```

次に、更新プログラム パッケージをコミットしたパッケージにロールバックする例を示します。

```
Device# install rollback to base

install_rollback: START Sat Feb 26 11:27:41 PST 2017
This rollback would require a reload. Do you want to proceed? [y/n]y
2 install_rollback: Reloading the box to take effect

Initializing Hardware ...
<after reload>
Device#
```

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

```
Device# show install summary

Active Packages:
bootflash:isr4300-
universalk9.BLD_POLARIS_DEV_SMU_LATEST_20170128_040557.1.CSCvbXXXXX.SSA.smu.bin
Inactive Packages:
No packages
Committed Packages:
bootflash:isr4300-
universalk9.BLD_POLARIS_DEV_SMU_LATEST_20170128_040557.1.CSCvbXXXXX.SSA.smu.bin
Uncommitted Packages:
No packages
Device#
```

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

```
Device# show install log
```

```
[0|install_op_boot]: START Sat Feb 26 19:31:50 Universal 2017
[0|install_op_boot]: END SUCCESS Sat Feb 26 19:31:56 Universal 2017
```

次に、SMU パッケージ ファイルを非アクティブ化する例を示します。

```
Device# install deactivate file bootflash:isr4300-
universalk9.BLD_POLARIS_DEV_SMU_LATEST_20170128_040557.1.CSCvbXXXXX.SSA.smu.bin

install_deactivate: START Sat Feb 26 10:49:07 PST 2017
The activation step would require a reload. Do you want to proceed? [y/n]y
Regular SMU. Reloading the box to complete activation of the SMU...

Initializing Hardware...
...
<after reload>
Device#
```

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

```
Device# show install summary

Active Packages:
No packages
Inactive Packages:
bootflash:isr4300-universalk9.BLD_POLARIS_DEV_SMU_LATEST_20170128_040557.1.CSCvbXXXXX.
SSA.smu.bin
Committed Packages:
No packages
Uncommitted Packages:
No packages
Device#
```

次に、デバイスから SMU を削除する例を示します。

```
Device# install remove file bootflash:isr4300-
universalk9.BLD_POLARIS_DEV_SMU_LATEST_20170128_040557.1.CSCvbXXXXX.SSA.smu.bin

install_remove: START Sat Feb 26 12:09:43 PST 2017
SUCCESS: install_remove /bootflash/isr4300-
universalk9.BLD_POLARIS_DEV_SMU_LATEST_20170128_040557.1.CSCvbXXXXX.SSA.smu.bin Sat Feb
26 12:09:49 PST 2017
Device#
```

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

```
Device# show install summary

Active Packages:
No packages
Inactive Packages:
No packages
Committed Packages:
No packages
Uncommitted Packages:
No packages
```

ソフトウェア メンテナンス アップグレードの機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースのみを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

プラットフォームのサポートおよびCiscoソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 *13 :* ソフトウェア メンテナンス アップグレードの機能情報

機能名	リリース	機能情報
ソフトウェア メンテナンス アップグレード	Cisco IOS XE Everest 16.6.1	SMU は、システムにインストールしてパッチ修正やセキュリティ解決をリリースされたイメージに提供するためのパッケージです。 この機能は、次のプラットフォームでサポートされます。 • Cisco Catalyst 9500 シリーズ スイッチ コマンド **install**、**show install** が導入または更新されました。

第 11 章

フラッシュ ファイル システムの操作

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。最新の機能情報および警告については、使用するプラットフォームおよびソフトウェア リリースの Bug Search Tool およびリリース ノートを参照してください。このモジュールに記載されている機能の詳細を検索し、各機能がサポートされているリリースのリストを確認する場合は、このモジュールの最後にある機能情報の表を参照してください。

プラットフォームのサポートおよびCisco ソフトウェア イメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator には、http://www.cisco.com/go/cfn からアクセスします。Cisco.com のアカウントは必要ありません。

フラッシュ ファイル システムについて

フラッシュ ファイル システムは、ファイルを格納できる単一のフラッシュ デバイスです。ソフトウェア バンドルおよびコンフィギュレーション ファイルの管理に役立つ複数のコマンドも備えています。デバイスのデフォルトのフラッシュ ファイル システムは flash: です。

アクティブなデバイスから見ると、flash: はローカル フラッシュ デバイスを指します。これは、ファイル システムが表示されているいるのと同じデバイスに接続されているデバイスです。

一度に 1 人のユーザのみが、ソフトウェア バンドルおよびコンフィギュレーション ファイルを管理できます。

使用可能なファイル システムの表示

デバイス で使用可能なファイル システムを表示するには、**show file systems** 特権 EXEC コマンドを使用します（次のスタンドアロン デバイス の例を参照）。

表 *14 : show file systems* のフィールドの説明

フィールド	値
Size(b)	ファイル システムのメモリ サイズ（バイト単位）です。
Free(b)	ファイル システムの空きメモリ サイズ（バイト単位）です。
タイプ（Type）	ファイル システムのタイプです。 **disk**：ファイル システムは、フラッシュ メモリ デバイス、USB フラッシュ、crashinfo ファイル用です。 **network**：ファイル システムは、FTP サーバや HTTP サーバなどのネットワーク デバイス用です。 **nvram**：ファイル システムは NVRAM（不揮発性 RAM）デバイス用です。 **opaque**：ファイルシステムは、ローカルに生成された pseudo ファイル システム（system など）、またはダウンロード インターフェイス（brimux など）です。 **unknown**：ファイル システムのタイプは不明です。
フラグ（Flags）	ファイル システムの権限です。 **ro**：読み取り専用です。 **rw**：読み取りおよび書き込みです。 **wo**：書き込み専用です。

フィールド	値
プレフィックス（Prefixes）	ファイル システムのエイリアスです。 **crashinfo**：crashinfo ファイルです。 **flash:** ：フラッシュ ファイル システムです。 **ftp**：FTP サーバです。 **http**：HTTP サーバです。 **https**：セキュア HTTP サーバです。 **nvram:** ：NVRAM です。 **null:** ：コピーのヌル宛先です。リモート ファイルをヌルへコピーして、サイズを判別できます。 **rcp**：Remote Copy Protocol（RCP）サーバです。 **scp**：Session Control Protocol（SCP）サーバです。 **system:** ：実行コンフィギュレーションを含むシステム メモリが格納されています。 **tftp:** ：TFTP ネットワーク サーバです。 **usbflash0**：USB フラッシュ メモリです。 **ymodem:** ：YMODEM プロトコルを使用して、ネットワーク マシンからファイルを取得します。

デフォルト ファイル システムの設定

デフォルトのファイル システムとして使用されるファイル システムまたはディレクトリを指定するには、**cd** *filesystem:* 特権 EXEC コマンドを使用します。デフォルト ファイル システムを設定すると、関連するコマンドを実行するときに *filesystem:* 引数を省略できます。たとえば、オプションの *filesystem:* 引数を持つすべての特権 EXEC コマンドでは、**cd** コマンドで指定されたファイル システムが使用されます。

デフォルトでは、デフォルト ファイル システムは *flash:* です。

cd コマンドで指定された現在のデフォルトのファイル システムを表示するには、**pwd** 特権 EXEC コマンドを使用します。

ファイル システムのファイルに関する情報の表示

ファイル システムの内容を操作する前に、そのリストを表示できます。たとえば、新しいコンフィギュレーション ファイルをフラッシュ メモリにコピーする前に、ファイル システムに同じ名前のコンフィギュレーション ファイルが格納されていないことを確認できます。同様に、

フラッシュ コンフィギュレーション ファイルを別の場所にコピーする前に、ファイル名を確認して、その名前を別のコマンドで使用できます。ファイルシステムのファイルに関する情報を表示するには、次の表に記載する特権 EXEC コマンドのいずれかを使用します。

表 *15*：ファイルに関する情報を表示するためのコマンド

コマンド	説明
dir [/all] [*filesystem:filename*]	ファイル システムのファイル リストを表示します。
show file systems	ファイル システムのファイルごとの詳細を表示します。
show file information *file-url*	特定のファイルに関する情報を表示します。
show file descriptors	開いているファイルの記述子のリストを表示します。ファイル記述子は開いているファイルの内部表現です。このコマンドを使用して、別のユーザによってファイルが開かれているかどうかを調べることができます。

たとえば、ファイル システムのすべてのファイルのリストを表示するには、次のように **dir** 特権 EXEC コマンドを使用します。

```
デバイス# dir flash:
DDirectory of bootflash:/

616513  drwx             4096  Jul 15 2015 07:11:35 +00:00  .installer
608402  -rw-            33818  Sep 25 2015 11:41:35 +00:00  bootloader_evt_handle.log
608403  drwx             4096  Feb 27 2017 13:56:47 +00:00  .ssh
608410  -rw-                0   Jun 5 2015 10:16:17 +00:00  dc_stats.txt
608411  drwx            20480  Sep 23 2015 11:50:13 +00:00  core
624625  drwx             4096  Sep 23 2015 12:29:27 +00:00  .prst_sync
640849  drwx             4096  Feb 27 2017 13:57:30 +00:00  .rollback_timer
608412  drwx             4096  Jun 17 2015 18:12:47 +00:00  orch_test_logs
608413  -rw-         33554432  Sep 25 2015 11:43:15 +00:00  nvram_config
608417  -rw-               35  Sep 25 2015 20:17:42 +00:00  pnp-tech-time
608439  -rw-           214054  Sep 25 2015 20:17:48 +00:00  pnp-tech-discovery-summary
608419  drwx             4096  Jul 23 2015 07:50:25 +00:00  util
616514  drwx             4096  Mar 18 2015 11:09:04 +00:00  onep
608442  -rw-              556  Mar 18 2015 11:19:34 +00:00  vlan.dat
608448  -rw-          1131779  Mar 28 2015 13:13:48 +00:00  log.txt
616516  drwx             4096   Apr 1 2015 09:34:56 +00:00  gs_script
616517  drwx             4096   Apr 6 2015 09:42:38 +00:00  tools
608440  -rw-              252  Sep 25 2015 11:41:52 +00:00  boothelper.log
624626  drwx             4096  Apr 17 2015 06:10:55 +00:00  SD_AVC_AUTO_CONFIG
608488  -rw-            98869  Sep 25 2015 11:42:15 +00:00  memleak.tcl
608437  -rwx            17866  Jul 16 2015 04:01:10 +00:00  ardbeg_x86
632745  drwx             4096  Aug 20 2015 11:35:09 +00:00  CRDU
632746  drwx             4096  Sep 16 2015 08:57:44 +00:00  ardmore
608418  -rw-          1595361   Jul 8 2015 11:18:33 +00:00
system-report_RP_0_20150708-111832-UTC.tar.gz
608491  -rw-         67587176  Aug 12 2015 05:30:35 +00:00  mcln_x86_kernel_20170628.SSA
608492  -rwx         74880100  Aug 12 2015 05:30:57 +00:00  stardust.x86.idprom.0718B

11250098176 bytes total (9128050688 bytes free)
デバイス#
```

ディレクトリの変更および作業ディレクトリの表示

ディレクトリを変更し、作業ディレクトリを表示するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**dir** *filesystem:* 例： `Device# dir flash:`	指定されたファイル システムのディレクトリを表示します。 *filesystem:* には、システム ボードのフラッシュ デバイスの flash: を使用します。
ステップ 3	**cd** *directory_name* 例： `Device# cd new_configs`	指定されたディレクトリへ移動します。 コマンド例では、*new_configs* という名前のディレクトリに移動する方法を示します。
ステップ 4	**pwd** 例： `Device# pwd`	作業ディレクトリを表示します。
ステップ 5	**cd** 例： `Device# cd`	デフォルトディレクトリに移動します。

ディレクトリの作成

特権 EXEC モードを開始して、ディレクトリを作成するには次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**dir** *filesystem:* 例： `Device# dir flash:`	指定されたファイル システムのディレクトリを表示します。 *filesystem:* には、システム ボードのフラッシュ デバイスの flash: を使用します。
ステップ 2	**mkdir** *directory_name* 例： `Device# mkdir new_configs`	新しいディレクトリを作成します。スラッシュ (/) 間に指定できるディレクトリ名は最大 45 文字で、大文字と小文字の区別があります。ディレクトリ名には制御文字、スペース、スラッシュ、引用符、セミコロン、またはコロンは使用できません。
ステップ 3	**dir** *filesystem:* 例： `Device# dir flash:`	入力を確認します。

ディレクトリの削除

ディレクトリを、その内部のすべてのファイルおよびサブディレクトリとともに削除するには、delete /force /recursive**delete /force /recursive**filesystem:/file-url 特権 EXEC コマンドを使用します。

名前で指定されたディレクトリを、その内部のすべてのサブディレクトリおよびファイルとともに削除するには、**/recursive** キーワードを使用します。ディレクトリ内のファイルごとに表示される、削除を確認するためのプロンプトを省略するには、**/force** キーワードを使用します。この削除プロセスを実行すると、最初に 1 度だけプロンプトが表示されます。

filesystem でシステム ボードのフラッシュ デバイスを指定する場合は、**flash:** を使用します。*file-url* には、削除するディレクトリの名前を入力します。ディレクトリ内のすべてのファイルおよびディレクトリが削除されます。

⚠

注意 ディレクトリが削除された場合、その内容は回復できません。

ファイルのコピー

送信元から宛先にファイルをコピーするには、**copy** *source-url destination-url* 特権 EXEC コマンドを使用します。送信元および宛先の URL には、**running-config** および **startup-config** キーワード ショートカットを使用できます。たとえば、**copy running-config startup-config** コマンドを実行すると、現在の実行コンフィギュレーション ファイルがフラッシュ メモリの NVRAM セクションに保存され、システム初期化中のコンフィギュレーションとして使用されます。

XMODEM または YMODEM プロトコルを使用するネットワーク マシンのファイルに対する送信元として特殊なファイル システム（**xmodem:**、**ymodem:**）を指定し、そこからコピーすることもできます。

ネットワーク ファイル システムの URL には、ftp: 、rcp: 、tftp などがあり、構文は次のとおりです。

- FTP：ftp:[[//username [:password]@location]/directory]/filename
- RCP：rcp:[[//username@location]/directory]/filename
- TFTP：tftp:[[//location]/directory]/filename

ローカルにある書き込み可能なファイル システムには flash: などがあります。

送信元および宛先の組み合わせによっては、無効な場合があります。特に、次に示す組み合わせの場合は、コピーできません。

- 実行コンフィギュレーションから実行コンフィギュレーションへ
- スタートアップ コンフィギュレーションからスタートアップ コンフィギュレーションへ
- デバイスから同じ名前のデバイスへ（たとえば、**copy flash: flash:** コマンドは無効）

スタック内のDeviceから同じスタックの別のDeviceにファイルをコピーする

スタック内のあるデバイスから同じスタック内の別のデバイスにファイルをコピーするには、**flash-X:** 表記を使用します。**X** はデバイス番号です。

スタック内のすべてのデバイスを表示するには、9 メンバーデバイス スタックの例のように、特権 EXEC モードで **show switch** コマンドを使用します。

特定のデバイスのコピー可能なすべてのファイル システムを表示するには、次に示す 5 メンバー スタックの例のように、**copy** コマンドを使用します。

次の例では、デバイス 2 のフラッシュ パーティションに保存されているコンフィギュレーション ファイルをデバイス 4 のフラッシュ パーティションにコピーする方法を示しています。デバイス 2 とデバイス 4 が同じスタック内にあるとします。

```
Device# copy flash-2:config.txt flash-4:config.txt
```

ファイルの削除

フラッシュ メモリ デバイスのファイルが不要になった場合は、そのファイルを永久に削除できます。指定されたフラッシュ デバイスからファイルまたはディレクトリを削除するには、**delete** [**/force**] [**/recursive**] [*filesystem:*]*/file-url* 特権 EXEC コマンドを使用します。

ディレクトリを、その内部のすべてのサブディレクトリやファイルとともに削除するには、**/recursive** キーワードを使用します。ディレクトリ内のファイルごとに表示される、削除を確認するためのプロンプトを省略するには、**/force** キーワードを使用します。この削除プロセスを実行すると、最初に 1 度だけプロンプトが表示されます。**/force** キーワードおよび **/recursive** キーワードを使用して、**archive download-sw** コマンドを使用してインストールされ、不要になった古いソフトウェア イメージを削除します。

filesystem: オプションを省略すると、デバイスは **cd** コマンドで指定したデフォルトのデバイスを使用します。*file-url* には、削除するファイルのパス（ディレクトリ）および名前を指定します。

ファイルを削除しようとすると、削除の確認を求めるプロンプトが表示されます。

注意 ファイルが削除された場合、その内容は回復できません。

ここでは、デフォルトのフラッシュ メモリ デバイスからファイル *myconfig* を削除する例を示します。

```
Device# delete myconfig
```

ファイルの作成、表示、および抽出

ファイルを作成してそこにファイルを書き込んだり、ファイル内のファイルをリスト表示したり、ファイルからファイルを抽出したりできます（次の項を参照）。

ファイルの作成、内容の表示、およびファイルの抽出を行うには、特権 EXEC コマンドで次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**archive tar /create** *destination-url***flash:** */file-url* **例：** `デバイス# archive tar /create tftp:172.20.10.30/saved. flash:/new-configs`	ファイルを作成し、そこにファイルを追加します。 destination-url には、ローカルまたはネットワーク ファイル システムの宛先 URL のエイリアス、および作成す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
		• ローカル フラッシュ ファイル システム構文 **flash:** • FTP 構文 **ftp**:[[//*username*[:*password*][@*location*]/*directory*]/*-filename*. • RCP 構文 **rcp**:[[//*username@location*]/*directory*]/*-filename*. • TFTP 構文 **tftp**:[[//*location*]/*directory*]/*-filename*. **flash:**/*file-url* には、ローカル フラッシュ ファイル システム上の、新しいファイルが作成される場所を指定します。送信元ディレクトリ内に格納されている任意のファイルまたはディレクトリの一覧を指定して、新しいファイルに追加することもできます。何も指定しないと、このレベルにおけるすべてのファイルおよびディレクトリが、新規に作成されたファイルに書き込まれます。
ステップ 2	**archive tar /table** *source-url* 例： `デバイス# archive tar /table flash: /new_configs`	ファイルの内容を表示します。 *source-url* には、ローカル ファイル システムまたはネットワーク ファイル システムの送信元 URL エイリアスを指定します。*-filename.* は、表示するファイルです。次のオプションがサポートされています。 • ローカル フラッシュ ファイル システム構文 **flash:** • FTP 構文 **ftp**:[[//*username*[:*password*][@*location*]/*directory*]/*-filename*. • RCP 構文 **rcp**:[[//*username@location*]/*directory*]/*-filename*. • TFTP 構文 **tftp**:[[//*location*]/*directory*]/*-filename*. ファイルのあとにファイルまたはディレクトリのリストを指定して、ファイルの表示を制限することもできます。指定し

	コマンドまたはアクション	目的
		たファイルだけが表示されます。何も指定しないと、すべてのファイルおよびディレクトリが表示されます。
ステップ 3	**archive tar /xtract** *source-url* **flash:**/*file-url* [*dir/file...*] 例： `デバイス# archive tar /xtract tftp:/172.20.10.30/saved. flash:/new-configs`	ファイルをフラッシュ ファイル システム上のディレクトリに抽出します。 *source-url* には、ローカル ファイル システムの送信元 URL のエイリアスを指定します。*-filename.* は、ファイルの抽出元のファイルです。次のオプションがサポートされています。 • ローカル フラッシュ ファイル システム構文 **flash:** • FTP 構文 **ftp**:[[//*username*[:*password*]@*location*]/*directory*]/*-filename*. • RCP 構文 **rcp**:[[//*username@location*]/*directory*]/*-filename*. • TFTP 構文 **tftp**:[[//*location*]/*directory*]/*-filename*. **flash:**/*file-url* [*dir/file...*] には、ファイルの抽出元にするローカル フラッシュ ファイル システム上の場所を指定します。抽出対象のファイル内のファイルまたはディレクトリのリストを指定するには、*dir/file...* オプションを使用します。何も指定されないと、すべてのファイルとディレクトリが抽出されます。
ステップ 4	**more** [**/ascii** | **/binary** | **/ebcdic**] /*file-url* 例： `デバイス# more flash:/new-configs`	リモート ファイル システム上のファイルを含めて、読み取り可能なファイルの内容を表示します。

フラッシュファイルシステムに関するその他の関連資料

関連資料

関連項目	参照先
flash: ファイル システムの管理コマンド	『*Cisco IOS Configuration Fundamentals Command Reference*』

標準

標準	タイトル
この機能によってサポートされる新しい標準または変更された標準はありません。またこの機能による既存標準のサポートに変更はありません。	--

MIB

MIB	MIB リンク
この機能によってサポートされる新しいMIBまたは変更されたMIBはありません。またこの機能による既存 MIB のサポートに変更はありません。	選択したプラットフォーム、Cisco IOS リリース、およびフィーチャ セットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

RFC

RFC	タイトル
この機能によりサポートされた新規 RFC または改訂 RFC はありません。またこの機能による既存 RFC のサポートに変更はありません。	--

テクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Product Alert Tool（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/cisco/web/support/index.html

フラッシュ ファイル システムの機能履歴と情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースのみを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

リリース	変更箇所
Cisco IOS XE Everest 16.5.1a	この機能が導入されました。

第 12 章

条件付きデバッグとラジオアクティブ トレース

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。最新の機能情報および警告については、使用するプラットフォームおよびソフトウェア リリースの Bug Search Tool およびリリース ノートを参照してください。このモジュールに記載されている機能の詳細を検索し、各機能がサポートされているリリースのリストを確認する場合は、このモジュールの最後にある機能情報の表を参照してください。

プラットフォームのサポートおよびCisco ソフトウェア イメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator には、http://www.cisco.com/go/cfn からアクセスします。Cisco.com のアカウントは必要ありません。

条件付きデバッグの概要

条件付きデバッグ機能によって、定義した条件に基づき、特定の機能のデバッグおよびロギングを選択して有効にすることができます。この機能は、多くの機能がサポートされているシステムで有用です。

（注） コントロール プレーン トレースのみがサポートされています。

条件付きデバッグでは、多数の機能が導入されていて大規模に稼働しているネットワークにおけるきめ細かなデバッグが可能です。これにより、システム内の細かなインスタンスに対しても、詳細なデバッグを実行できます。これは、何千ものセッションのうちの特定のセッションのみをデバッグするような場合に、非常に有用です。条件は複数指定することもできます。

条件とは、機能またはアイデンティティをいいます。アイデンティティは、インターフェイス、IP アドレス、MAC アドレスなどです。

（注） サポートされる条件は MAC アドレスであることのみです。

これは、処理する機能オブジェクトを区別せずに出力を生成する、一般的なデバッグ コマンドとは対照的です。一般的なデバッグ コマンドは、多数のシステム リソースを消費し、システム パフォーマンスに影響します。

ラジオアクティブ トレースの概要

ラジオアクティブ トレースにより、冗長性のレベルを高めた状態で、システムの全体にわたって目的とする動作を連鎖的に実行できます。また、複数のスレッド、プロセス、および関数呼び出しにわたって、デバッグ情報を条件に基づいて（DEBUG レベルまで、または指定のレベルまで）出力する方法を提供します。

（注） デフォルトのレベルは **DEBUG** です。ユーザは別のレベルに変更することはできません。

条件付きデバッグとラジオアクティブトレースの設定方法

条件付きデバッグおよび放射線トレース

条件付きデバッグと組み合わせた放射線トレースによって、条件に関連するすべての実行コンテキストをデバッグする単一のデバッグ CLI を取得できます。これは、ボックス内の機能のさまざまな制御フロー プロセスを認識していなくても行うことができ、これらのプロセスでデバッグを個別に発行する必要もありません。

トレースファイルの場所

デフォルトでは、トレースファイル ログは各プロセスで生成され、**/tmp/rp/trace** または **/tmp/fp/trace** ディレクトリに保存されます。この一時ディレクトリで、トレース ログがファイルに書き込まれます。各ファイルは 1 MB サイズです。このディレクトリでは、特定のプロセスのこうしたファイルを、最大 25 件保持できます。**/tmp** ディレクトリのトレースファイルがその 1 MB 制限またはブート時に設定されたサイズに達した場合、ローテーションから外れ、**tracelogs** ディレクトリの **/crashinfo** パーティションの下にあるアーカイブの場所に移動します。

/tmp ディレクトリが 1 つのプロセスで保持するトレースファイルは 1 つのみです。ファイルがそのファイル サイズの制限に達したら、ローテーションから外れ、**/crashinfo/tracelogs** に移動します。アーカイブ ディレクトリに蓄積されるファイルは最大 25 ファイルであり、その後は最も古いものから順に、**/tmp** から新たにローテーションされたファイルに置換されます。

crashinfo ディレクトリ内のトレースファイルは次の形式で配置されます。

1. Process-name_Process-ID_running-counter.timestamp.gz

   例：IOSRP_R0-0.bin_0.14239.20151101234827.gz

2. Process-name_pmanlog_Process-ID_running-counter.timestamp.bin.gz

   例：wcm_pmanlog_R0-0.30360_0.20151028233007.bin.gz

条件付きデバッグの設定

条件付デバッグを設定するには、以下の手順に従います。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**debug platform condition mac** {*mac-address*} 例： `Device# debug platform condition mac bc16.6509.3314`	指定された MAC アドレスの条件付きデバッグを設定します。
ステップ 3	**debug platform condition start** 例： `Device# debug platform condition start`	条件付きデバッグを開始します（上記のいずれかの条件に一致すると放射線トレースを開始します）。

	コマンドまたはアクション	目的

ステップ 4	**show platform condition** または **show debug**
	例：
	`Device# show platform condition`
	`Device# show debug`
		現在設定されている条件を表示します。

ステップ 5	**debug platform condition stop** 例： `Device# debug platform condition stop`	条件付きデバッグを停止します（放射線トレースを停止します）。
ステップ 6	**request platform software trace archive** [**last** {*number*} **days**] [**target** {*crashinfo:* | *flashinfo:*}] 例： `Device# request platform software trace archive last 2 days`	（任意）システムのマージされたトレースファイルの履歴ログを表示します。日数またはロケーションの組み合わせのフィルタ。
ステップ 7	**show platform software trace** [**filter-binary** | **level** | **message**] 例： `Device# show platform software trace message`	（任意）最新のトレースファイルからマージされたログを表示します。アプリケーションの状態、トレース モジュール名およびトレース レベルをさまざまな組み合わせでフィルタリングします。 • **filter-binary** ：照合するモジュールをフィルタリングします。 • **level** ：トレース レベルを表示します。 • **message** ：トレース メッセージのリングの内容を表示します。 （注） デバイス上では次が可能です。 • Linux シェルだけでなく、IOS のコンソールからも使用できます。 • マージされたログでファイルを生成します。 • ステージング エリアからのみマージされたログを表示します。

	コマンドまたはアクション	目的
ステップ 8	**clear platform condition all** 例： `Device# clear platform condition all`	すべての条件をクリアします。

次のタスク

（注） **request platform software trace filter-binary** および **show platform software trace filter-binary** コマンドは、似たように動作します。唯一の違いは次のとおりです。

- **request platform software trace filter-binary** ：データ ソースとして履歴ログを使用します。
- **show platform software trace filter-binary** ：データ ソースとしてフラッシュの一時ディレクトリを使用します。

その中でも、*mac_log <..date..>* は、デバッグする MAC 用のメッセージを伝えるため、最も重要なファイルです。**show platform software trace filter-binary** コマンドも同じフラッシュ ファイルを生成し、また、画面に mac_log を出力します。

L2 マルチキャストの放射線トレース

特定のマルチキャスト受信者を特定するには、参加者または受信側クライアントの MAC アドレス、グループのマルチキャスト IP アドレスおよびスヌーピング VLAN を指定します。また、デバッグのトレース レベルを有効にします。デバッグ レベルでは、詳細なトレースとシステムへの高い可視性が提供されます。

debug platform condition feature multicast controlplane mac *client MAC address* **ip** *Group IP address* **vlan** *id* **level** *debug level*

トレース ファイルの推奨ワークフロー

トレース ファイルの推奨ワークフローの概要は次のとおりです。

1. 特定の時間帯のトレースログを要求する場合。

   たとえば 1 日。

   使用するコマンドは、次のとおりです。

   Device#**リクエストプラットフォームソフトウェアトレースアーカイブ過去 1 日間**

2. システムは、/flash: ロケーション内のトレースログの tar ball（.gz ファイル）を生成します。

3. スイッチ外にファイルをコピーします。ファイルをコピーすることによって、オフラインでトレースログが使用できます。ファイルのコピーについての詳細は、次のセクションを参照してください。
4. /flash: location からトレースログファイル（.gz）ファイルを削除します。これにより、他の操作に十分な領域がスイッチに確保されます。

ボックス外へのトレース ファイルのコピー

トレース ファイルの例を以下に示します。

```
Device# dir crashinfo:/tracelogs
Directory of crashinfo:/tracelogs/

50664 -rwx 760 Sep 22 2015 11:12:21 +00:00 plogd_F0-0.bin_0.gz
50603 -rwx 991 Sep 22 2015 11:12:08 +00:00 fed_pmanlog_F0-0.bin_0.9558.20150922111208.gz
50610 -rw- 11 Nov 2 2015 00:15:59 +00:00 timestamp
50611 -rwx 1443 Sep 22 2015 11:11:31 +00:00
auto_upgrade_client_sh_pmanlog_R0-.bin_0.3817.20150922111130.gz
50669 -rwx 589 Sep 30 2015 03:59:04 +00:00 cfgwr-8021_R0-0.bin_0.gz
50612 -rwx 1136 Sep 22 2015 11:11:46 +00:00 reflector_803_R0-0.bin_0.1312.20150922111116.gz
50794 -rwx 4239 Nov 2 2015 00:04:32 +00:00 IOSRP_R0-0.bin_0.14239.20151101234827.gz
50615 -rwx 131072 Nov 2 2015 00:19:59 +00:00 linux_iosd_image_pmanlog_R0-0.bin_0
--More—
```

トレース ファイルは、次に示すさまざまなオプションのいずれかを使用して、コピーできます。

```
Device# copy crashinfo:/tracelogs ?
  crashinfo: Copy to crashinfo: file system
  flash: Copy to flash: file system
  ftp: Copy to ftp: file system
  http: Copy to http: file system
  https: Copy to https: file system
  null: Copy to null: file system
  nvram: Copy to nvram: file system
  rcp: Copy to rcp: file system
  running-config Update (merge with) current system configuration
  scp: Copy to scp: file system
  startup-config Copy to startup configuration
  syslog: Copy to syslog: file system
  system: Copy to system: file system
  tftp: Copy to tftp: file system
  tmpsys: Copy to tmpsys: file system
```

TFTP サーバにコピーするための一般的な構文は次のとおりです。

```
Device# copy source: tftp:
Device# copy crashinfo:/tracelogs/IOSRP_R0-0.bin_0.14239.20151101234827.gz tftp:
Address or name of remote host []? 2.2.2.2
Destination filename [IOSRP_R0-0.bin_0.14239.20151101234827.gz]?
```

（注） tracelogおよび他の目的に使用可能な空き容量があることを確認するために、生成されたレポート/アーカイブ ファイルをスイッチからクリアすることが重要です。

条件付きデバッグのモニタリング

以下の表に、条件付きデバッグのモニタに使用できる各種コマンドを示します。

コマンド	目的
show platform condition	現在設定されている条件を表示します。
show debug	現在設定されているデバッグ条件を表示します。
show platform software trace filter-binary	最新のトレース ファイルからマージされたログを表示します。
request platform software trace filter-binary	システムにマージされたトレース ファイルの履歴ログを表示します。

条件付きデバッグの設定例

次に、*show platform condition* コマンドの出力例を示します。

```
Device# show platform condition
Conditional Debug Global State: Stop
Conditions Direction
-----------------------------------------------------------------------------------------------|---------
MAC Address 0024.D7C7.0054 N/A
Feature Condition Type Value
-----------------------|-----------------------|---------------------------------
Device#
```

次に、*show debug* コマンドの出力例を示します。

```
Device# show debug
IOSXE Conditional Debug Configs:
Conditional Debug Global State: Start
Conditions Direction
-----------------------------------------------------------------------------------------------|---------
MAC Address 0024.D7C7.0054 N/A
Feature Condition Type Value
-----------------------|-----------------------|---------------------------------
Packet Infra debugs:
Ip Address Port
-------------------------------------------------------|-----------
Device#
```

次に、*debug platform condition stop* コマンドの例を示します。

```
Device# debug platform condition stop
Conditional Debug Global State: Stop
```

条件付きデバッグとラジオアクティブトレースに関するその他の関連資料

関連資料

関連項目	参照先
この章で使用するコマンドの完全な構文および使用方法の詳細。	*Command Reference (Catalyst 9500 Series Switches)*

MIB

MIB	MIB リンク
本リリースでサポートするすべての MIB	選択したプラットフォーム、Cisco IOS リリース、およびフィーチャ セットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

テクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Product Alert Tool（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

条件付きデバッグとラジオアクティブトレースの機能履歴と情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースの

みを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

リリース	変更箇所
Cisco IOS XE Everest 16.5.1a	この機能が導入されました。

第 13 章

ソフトウェア設定のトラブルシューティング

この章では、スイッチが稼働する Cisco IOS ソフトウェアに関連する問題を特定し、解決する方法について説明します。問題の性質に応じて、コマンドラインインターフェイス（CLI）、デバイス マネージャ、または Network Assistant を使用して、問題を特定し解決できます。

LED の説明など、トラブルシューティングの詳細については、ハードウェアインストレーション ガイドを参照してください。

ソフトウェア設定のトラブルシューティングに関する情報

スイッチのソフトウェア障害

スイッチ ソフトウェアがアップグレード中に破損する原因として、誤ったファイルがスイッチにダウンロードされた場合やイメージファイルが削除された場合があります。これらのどの場合も、スイッチは、電源投入時自己診断テスト（POST）に合格せず、接続はありません。ソフトウェア障害から回復するには、ソフトウェア障害からの回復 （263 ページ）の項で説明されている手順に従います。

のパスワードを紛失したか忘れた場合 デバイス

デバイスのデフォルト設定では、デバイスに物理的にアクセスしているエンドエンド ユーザは、スイッチの電源投入中に起動プロセスを中断して新しいパスワードを入力することにより、パスワードを紛失した状態から回復できます。ここで紹介する回復手順を実行するには、デバイスに物理的にアクセスする必要があります。

（注） これらのデバイスでは、システム管理者は、デフォルト設定に戻すことに同意した場合に限り、エンド ユーザによるパスワードのリセットを許可することによって、この機能の一部をディセーブルにできます。パスワード回復がディセーブルになっている場合に、エンドユーザがパスワードをリセットしようとすると、ステータスメッセージで回復プロセスの間はデフォルトの設定に戻すように指示されます。

（注） Cisco WLC の設定を複数の Cisco WLC 間でコピーすると、暗号化パスワード キーを回復できなくなります（RMA の場合）。

パスワードを紛失または忘れた場合にそのパスワードを回復するには、パスワードを忘れた場合の回復 （264 ページ）の項で説明する手順に従います。

Ping

デバイスは IP の ping をサポートしており、これを使ってリモート ホストへの接続をテストできます。ping はアドレスにエコー要求パケットを送信し、応答を待ちます。ping は次のいずれかの応答を返します。

- 正常な応答：正常な応答（*hostname* が存在する）は、ネットワーク トラフィックにもよりますが、1 ～ 10 秒以内で発生します。
- 宛先の応答なし：ホストが応答しない場合、*no-answer* メッセージが返されます。
- 不明なホスト：ホストが存在しない場合、*unknown host* メッセージが返されます。
- 宛先到達不能：デフォルト ゲートウェイが指定されたネットワークに到達できない場合、*destination-unreachable* メッセージが返されます。
- ネットワークまたはホストへの到達不能：ルート テーブルにホストまたはネットワークのエントリがない場合、*network or host unreachable* メッセージが返されます。

ping の動作を理解するには、ping の実行 （270 ページ）の項を参照してください。

レイヤ 2 Traceroute

レイヤ 2 traceroute 機能により、パケットが通過する、送信元デバイスから宛先デバイスへの物理パスを識別できます。レイヤ 2 traceroute は、ユニキャストの送信元および宛先 MAC アドレ

スだけをサポートします。transroute は、パス内にあるデバイスの MAC アドレス テーブルを使用してパスを識別します。デバイスがパス内でレイヤ 2 traceroute をサポートしていないデバイスを検知した場合、デバイスはレイヤ 2 trace クエリーを送信し続け、タイムアウトにします。

デバイスは、送信元デバイスから宛先デバイスへのパスのみを識別できます。パケットが通過する、送信元ホストから送信元デバイスまで、または宛先デバイスから宛先ホストまでのパスは識別できません。

レイヤ 2 の traceroute のガイドライン

- ネットワーク内のすべてのデバイスで、Cisco Discovery Protocol（CDP）をイネーブルにする必要があります。レイヤ 2 traceroute が適切に動作するために、CDP をディセーブルにしないでください。

  物理パス内のデバイスが CDP に対して透過的な場合、スイッチはこれらのデバイスを通過するパスを識別できません。

- **ping** 特権 EXEC コマンドを使用して接続をテストできれば、このデバイスは別のデバイスから到達可能といえます。物理パス内のすべてのデバイスは、他のスイッチから相互に到達可能でなければなりません。

- パス内で識別可能な最大ホップ数は 10 です。

- 送信元デバイスと宛先デバイスの間の物理パス内にないデバイスで、**traceroute mac** または **traceroute mac ip** の特権 EXEC コマンドを実行できます。パス内のすべてのデバイスは、このスイッチから到達可能でなければなりません。

- **traceroute mac** コマンドの出力結果としてレイヤ 2 パスが表示されるのは、指定の送信元および宛先 MAC アドレスが、同一の VLAN に属している場合だけです。指定した送信元および宛先 MAC アドレスが、それぞれ異なる VLAN に属している場合は、レイヤ 2 パスは識別されず、エラー メッセージが表示されます。

- マルチキャストの送信元または宛先 MAC アドレスを指定すると、パスは識別されず、エラー メッセージが表示されます。

- 送信元または宛先 MAC アドレスが複数の VLAN に属する場合は、送信元および宛先 MAC アドレスの両方が属している VLAN を指定する必要があります。VLAN を指定しないと、パスは識別されず、エラー メッセージが表示されます。

- 指定した送信元および宛先の IP アドレスが同一サブネットに属する場合、**traceroute mac ip** コマンド出力にレイヤ 2 パスが表示されます。IP アドレスを指定した場合、デバイスは Address Resolution Protocol（ARP）を使用し、IP アドレスとそれに対応する MAC アドレスおよび VLAN ID を関連付けます。

  - 指定の IP アドレスに ARP のエントリが存在している場合、デバイスは関連する MAC アドレスを使用して、物理パスを識別します。

  - ARP のエントリが存在しない場合、デバイスは ARP クエリを送信し、IP アドレスの解決を試みます。IP アドレスが解決されない場合は、パスは識別されず、エラーメッセージが表示されます。

- 複数のデバイスがハブを介して1つのポートに接続されている場合（たとえば複数のCDPネイバーがポートで検出された場合）、レイヤ2 traceroute機能はサポートされません。複数のCDPネイバーが1つのポートで検出された場合、レイヤ2パスは特定されず、エラーメッセージが表示されます。
- この機能は、トークンリング VLAN ではサポートされません。

IP Traceroute

IP traceroute を使用すると、ネットワーク上でパケットが通過するパスをホップバイホップで識別できます。このコマンドを実行すると、トラフィックが宛先に到達するまでに通過するルータなどのすべてのネットワーク層（レイヤ 3）デバイスが表示されます。

デバイスは、**traceroute** 特権 EXEC コマンドの送信元または宛先として指定できます。また、スイッチは **traceroute** コマンドの出力でホップとして表示される場合があります。デバイスを traceroute の宛先とすると、スイッチは、traceroute の出力で最終の宛先として表示されます。中間デバイスが同じVLAN内でポート間のパケットのブリッジングだけを行う場合、traceroute の出力に中間スイッチは表示されません。ただし、中間デバイスが、特定のパケットをルーティングするマルチレイヤ デバイスの場合、中間デバイスは traceroute の出力にホップとして表示されます。

traceroute 特権 EXEC コマンドは、IP ヘッダーの存続可能時間（TTL）フィールドを使用して、ルータおよびサーバで特定のリターン メッセージが生成されるようにします。traceroute の実行は、ユーザ データグラム プロトコル（UDP）データグラムを、TTL フィールドが 1 に設定されている宛先ホストへ送信することから始まります。ルータで TTL 値が 1 または 0 であることを検出すると、データグラムをドロップし、インターネット制御メッセージプロトコル（ICMP）time-to-live-exceeded メッセージを送信元に送信します。traceroute は、ICMP time-to-live-exceeded メッセージの送信元アドレス フィールドを調べて、最初のホップのアドレスを判別します。

ネクスト ホップを識別するために、traceroute は TTL 値が 2 の UDP パケットを送信します。1 番めのルータは、TTL フィールドの値から 1 を差し引いて次のルータにデータグラムを送信します。2 番めのルータは、TTL 値が 1 であることを確認すると、このデータグラムを廃棄し、time-to-live-exceeded メッセージを送信元へ返します。このように、データグラムが宛先ホストに到達するまで（または TTL の最大値に達するまで）TTL の値は増分され、処理が続けられます。

データグラムが宛先に到達したことを学習するために、traceroute は、データグラムの UDP 宛先ポート番号を、宛先ホストが使用する可能性のない大きな値に設定します。ホストが、ローカルで使用されない宛先ポート番号を持つ自分自身宛てのデータグラムを受信すると、送信元に ICMP ポート到達不能エラーを送信します。ポート到達不能エラーを除くすべてのエラーは中間ホップから送信されるため、ポート到達不能エラーを受信するということは、このメッセージが宛先ポートから送信されたことを意味します。

例：IP ホストに対する traceroute の実行 （282 ページ）に進み、IP traceroute プロセスの例を参照してください。

debug コマンド

⚠

注意 デバッグ出力は CPU プロセスで高プライオリティが割り当てられているため、デバッグ出力を行うとシステムが使用できなくなることがあります。したがって、**debug** コマンドを使用するのは、特定の問題のトラブルシューティング時、またはシスコのテクニカルサポート担当者とともにトラブルシューティングを行う場合に限定してください。ネットワーク トラフィック量やユーザ数が少ない期間に **debug** コマンドを使用することをお勧めします。デバッグをこのような時間帯に行うと、**debug** コマンド処理のオーバーヘッドの増加によりシステムの使用に影響が及ぶ可能性が少なくなります。

debug コマンドはすべて特権 EXEC モードで実行します。ほとんどの **debug** コマンドは引数を取りません。

システム レポート

システム レポートまたは crashinfo ファイルには、シスコのテクニカル サポート担当者が Cisco IOS イメージの障害（クラッシュ）が原因で起きた問題をデバッグするときに使用する情報が保存されています。明瞭度と整合性の高い重要なクラッシュ情報を迅速かつ確実に収集することが必要です。さらに、この情報の収集とバンドルが、特定のクラッシュの発生に対し関連付けか特定ができるような方法で行われることが必要です。

システム レポートは次の状況で生成されます。

- スイッチ障害の場合：システム レポートは障害が発生したスイッチで生成されます。
- スイッチオーバーの場合：システム レポートはハイ アベイラビリティ（HA）のメンバースイッチでのみ生成されます。非 HA メンバーについてはレポートは生成されません。

リロード時はレポートは生成されません。

クラッシュ プロセス時は、次の情報がスイッチからローカルに収集されます。

1. 完全なプロセス core
2. トレースログ
3. IOS の syslog（非アクティブなクラッシュの場合には保証されません）
4. システム プロセス情報
5. ブートアップ ログ
6. リロード ログ
7. 特定のタイプの /proc 情報

この情報は個別のファイルに格納されてから、アーカイブされて 1 つのバンドルに圧縮されます。これにより、クラッシュのスナップショットを 1 つの場所で取得して、分析のためにボッ

クス外に移動できるようになります。このレポートは、スイッチが ROMmon/ブートローダにダウンする前に生成されます。

完全な core およびトレースログ以外はテキスト ファイルです。

request platform software process core fed active コマンドを使用してコア ダンプを生成します。

```
h2-macallan1# request platform software process core fed active
Process : fed main event (28155) encountered fatal signal 6
Process : fed main event stack :

SUCCESS: Core file generated.

h2-macallan1#dir bootflash:core
Directory of bootflash:/core/

178483  -rw-                1  May 23 2017 06:05:17 +00:00  .callhome
194710  drwx             4096  Aug 16 2017 19:42:33 +00:00  modules
178494  -rw-         10829893  Aug 23 2017 09:46:23 +00:00
h2-macallan1_RP_0_fed_28155_20170823-094616-UTC.core.gz
```

crashinfo ファイル

デフォルトでは、生成されたシステム レポート ファイルは /crashinfo ディレクトリに格納されます。Ifit は、領域不足のため crashinfo パーティションに保存できません。そのため、/flash ディレクトリに保存されます。

ファイルを表示するには、**dir crashinfo:** コマンドを入力します。次に crashinfo ディレクトリの出力例を示します。

```
Switch#dir crashinfo:
Directory of crashinfo:/


23665 drwx 86016 Jun 9 2017 07:47:51 -07:00 tracelogs
11 -rw- 0 May 26 2017 15:32:44 -07:00 koops.dat
12 -rw- 4782675 May 29 2017 15:47:16 -07:00 system-report_1_20170529-154715-PDT.tar.gz
1651507200 bytes total (1519386624 bytes free)
```

システム レポートは、次の形式で crashinfo ディレクトリにあります。

```
system-report_[switch number]_[date]-[timestamp]-UTC.gz
```

スイッチがクラッシュしたら、システム レポート ファイルを確認します。最後に生成されたシステム レポート ファイルは crashinfo ディレクトリの下に last_systemreport というファイル名で保存されます。問題のトラブルシューティングを行う際、システム レポートおよびcrashinfo ファイルが TAC の役に立ちます。

生成されたシステム レポートは、TFTP や HTTP などいくつかのオプションを使用して、さらにコピーできます。

```
Switch#copy crashinfo: ?
crashinfo:     Copy to crashinfo: file system
flash:         Copy to flash: file system
ftp:           Copy to ftp: file system
http:          Copy to http: file system
https:         Copy to https: file system
null:          Copy to null: file system
nvram:         Copy to nvram: file system
rcp:           Copy to rcp: file system
running-config Update (merge with) current system configuration
```

```
scp:           Copy to scp: file system
startup-config Copy to startup configuration
syslog:        Copy to syslog: file system
system:        Copy to system: file system
tftp:          Copy to tftp: file system
tmpsys:        Copy to tmpsys: file system
```

TFTP サーバにコピーするための一般的な構文は次のとおりです。

```
Switch#copy crashinfo: tftp:
Source filename [system-report_1_20150909-092728-UTC.gz]?
Address or name of remote host []? 1.1.1.1
Destination filename [system-report_1_20150909-092728-UTC.gz]?
```

のトレースログは、trace archive コマンドを発行することで収集できます。このコマンドには、時間帯オプションがあります。コマンド構文は次のとおりです。

```
Switch#request platform software trace archive ?
last     Archive trace files of last x days
target   Location and name for the archive file
```

crashinfo: または flash: ディレクトリに格納されている過去 3650 日以内のトレースログが取得できます。

```
Switch# request platform software trace archive last ?
<1-3650> Number of days (1-3650)
Switch#request platform software trace archive last 3650 days target ?
crashinfo: Archive file name and location
flash:     Archive file name and location
```

（注） 一度コピーされたら、システム レポートやトレースのアーカイブを flash ディレクトリまたは crashinfo ディレクトリからクリアし、トレースログやその他の目的に使用できる領域を確保することが重要です。

スイッチのオンボード障害ロギング

オンボード障害ロギング（OBFL）機能を使用すれば、デバイスに関する情報を収集できます。この情報には稼働時間、温度、電圧などの情報が含まれており、シスコのテクニカルサポート担当者がデバイスの問題をトラブルシューティングする際に役立ちます。OBFL はイネーブルにしておき、フラッシュ メモリに保存されたデータは消さないようにすることを推奨します。

OBFL は、デフォルトでイネーブルになっています。デバイスおよび Small Form-Factor Pluggable（SFP）モジュールに関する情報が収集されます。デバイスは、次の情報をフラッシュ メモリに保存します。

- CLI コマンド：スタンドアロン デバイスに入力された OBFL CLI コマンドの記録
- 環境データ：スタンドアロン デバイスおよび接続されているすべての FRU デバイスの一意のデバイス ID（UDI）情報、製品 ID（PID）、バージョン ID（VID）、およびシリアル番号
- メッセージ：スタンドアロンデバイスにより生成されたハードウェア関連のシステム メッセージの記録

- イーサネット経由の電源供給（PoE）：スタンドアロンデバイスまたはのPoEポートの消費電力の記録
- 温度：スタンドアロン デバイスの温度
- 稼働時間：スタンドアロンデバイスが起動されたときの時刻、デバイスが再起動された理由、およびデバイスが最後に再起動されて以来の稼働時間
- 電圧：スタンドアロン デバイスのシステム電圧

システム時計は、手動で時刻を設定するか、またはネットワーク タイム プロトコル（NTP）を使用するように設定します。

デバイスの稼働中には、**show logging onboard** 特権 EXEC コマンドを使用することにより、OBFL データを取得できます。デバイスに障害が発生した場合のデータの取得方法については、お客様担当のシスコ テクニカル サポート担当者にお問い合わせください。

OBFL がイネーブルになっているデバイスが再起動された場合、新しいデータの記録が開始するまでに 10 分間の遅延があります。

ファン障害

デフォルトでは、この機能はディセーブルです。現場交換可能ユニット（FRU）または電源装置の複数のファンが故障した場合、デバイスはシャットダウンせず、次のようなエラー メッセージが表示されます。

```
Multiple fan(FRU/PS) failure detected. System may get overheated. Change fan quickly.
```

デバイスが過熱状態となり、シャットダウンすることもあります。

ファン障害機能をイネーブルにするには、**system env fan-fail-action shut** 特権 EXEC コマンドを入力します。デバイス内の複数のファンに障害が発生した場合、デバイスは自動的にシャットダウンし、次のようなエラー メッセージが表示されます。

```
Faulty (FRU/PS) fans detected, shutting down system!
```

最初のファンの停止後、デバイスが 2 つめのファンの障害を検知すると、デバイスは 20 秒待機してからシャットダウンします。

デバイスを再起動するには、電源をオフにしてから再度オンにする必要があります。

ファンの障害の詳細については、『Cisco Catalyst 9400 Series Switches Hardware Installaion Guide』を参照してください。

CPU 使用率が高い場合に起こりうる症状

CPU使用率が高すぎることで次の現象が発生する可能性がありますが、他の原因で発生する場合もあります。次にその一部を示します。

- スパニングツリー トポロジの変更
- 通信が切断されたために EtherChannel リンクがダウンした
- 管理要求（ICMP ping、SNMP のタイムアウト、低速な Telnet または SSH セッション）に応答できない
- UDLD フラッピング
- SLA の応答が許容可能なしきい値を超えたことによる IP SLA の失敗
- スイッチが要求を転送しない、または要求に応答しない場合の DHCP または IEEE 802.1x の処理の失敗

ソフトウェア設定のトラブルシューティング方法

ソフトウェア障害からの回復

始める前に

ここで紹介する回復手順を実行するには、スイッチを直接操作する必要があります。

ここで紹介する手順では、破損したイメージ ファイルまたは不適切なイメージ ファイルの回復に boot loader コマンドおよび TFTP を使用します。

手順

ステップ 1 PC 上で、Cisco.com からソフトウェア イメージファイル（*image.bin*）をダウンロードします。

ステップ 2 TFTP サーバにソフトウェア イメージをロードします。

ステップ 3 PC をスイッチのイーサネット管理ポートに接続します。

ステップ 4 スイッチの電源コードを取り外します。

ステップ 5 **Mode** ボタンを押しながら、電源コードをスイッチに再接続します。

例：

```
Last reset cause: SoftwareResetTrig
C9400-SUP-1 platform with 16777216 Kbytes of main memory

Preparing to autoboot. [Press Ctrl-C to interrupt]  3    (interrupted)
switch:
switch:
```

ステップ 6 ブートローダ（ROMMON）プロンプトで、TFTP サーバに ping を実行できることを確認します。

a) 次のコマンドを実行して、IP アドレス を設定します。**switch: set IP_ADDRESS** *ip_address subnet_mask*

例：

```
switch: set IP_ADDRESS 192.0.2.123/255.255.255.0
```

b) 次のコマンドを実行して、デフォルト ルータ IP アドレスを設定します。**switch: set DEFAULT_ROUTER** *ip_address*

例：

```
switch: set DEFAULT_ROUTER 192.0.2.1
```

c) 次のコマンドを実行して、TFTP サーバに ping を実行できることを確認します。**switch: ping** *ip_address_of_TFTP_server*

例：

```
switch: ping 192.0.2.15
ping 192.0.2.1 with 32 bytes of data...
Host 192.0.2.1 is alive.
switch:
```

ステップ 7 回復パーティション（sda9:）に回復イメージが存在することを確認します。

この回復イメージは、emergency-install機能を使用して回復を実施する場合に必要となります。

例：

```
switch: dir sda9:
Directory of sda9:/

    2  drwx  1024       .
    2  drwx  1024       ..
   11  -rw-  18923068   c3850-recovery.bin

36939776 bytes available (20830208 bytes used)
switch:
```

ステップ 8 ブートローダ（ROMMON）プロンプトで、emergency-install機能を開始します。これにより、スイッチでソフトウェア イメージを容易に回復できます。

警告：emergency-install コマンドを実行すると、ブート ブラッシュ全体が消去されます。

あるいは、Telnet または管理ポートを通じて TFTP からローカル フラッシュにイメージをコピーした後、ローカル フラッシュからデバイスをブートします。

パスワードを忘れた場合の回復

スイッチのデフォルト設定では、スイッチを直接操作するエンドユーザが、スイッチの電源投入時に起動プロセスを中断して新しいパスワードを入力することにより、パスワードを紛失した状態から回復できます。ここで紹介する回復手順を実行するには、スイッチを直接操作してください。

（注） これらのスイッチでは、システム管理者はデフォルト設定に戻す場合に限りエンドユーザによるパスワードのリセットを許可することによって、この機能の一部をディセーブルにできます。パスワード回復がディセーブルになっている場合に、エンドユーザがパスワードをリセットしようとすると、回復プロセスの間、ステータス メッセージにその旨が表示されます。

手順

ステップ 1 端末または PC をスイッチに接続します。

- 端末または端末エミュレーション ソフトウェアが稼働している PC をスイッチのコンソール ポートに接続します。
- PC をイーサネット管理ポートに接続します。

ステップ 2 エミュレーション ソフトウェアの回線速度を 9600 ボーに設定します。

ステップ 3 スタンドアロン スイッチまたはスイッチ スタック全体の電源を切断します。

ステップ 4 電源コードまたはアクティブ スイッチを再度接続します。15 秒以内に [**Mode**] ボタンを押します。このときシステム LED はグリーンに点滅しています。プロンプトが表示されるまで [**Mode**] ボタンを押し続けます。プロンプトが表示されたら [**Mode**] ボタンを放します。

```
Switch:
Base ethernet MAC Address: 20:37:06:4d:e9:80
Verifying bootloader digital signature.


The system has been interrupted prior to loading the operating
system software, console will be reset to 9600 baud rate.
```

「パスワード回復がイネーブルになっている場合の手順」セクションに記載されている手順を実行します。

ステップ 5 パスワードの回復後、スイッチまたは アクティブ スイッチ をリロードします。

スイッチの場合

```
Switch> reload
Proceed with reload? [confirm] y
```

パスワード回復がイネーブルになっている場合の手順

手順

ステップ 1 次のコマンドを使用して、スタートアップ コンフィギュレーションを無視します。

```
Switch: SWITCH_IGNORE_STARTUP_CFG=1
```

ステップ 2 *packages.conf* ファイルでスイッチをフラッシュからブートします。

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

ステップ 3 **No** と応答して初期設定ダイアログを終了します。

```
Would you like to enter the initial configuration dialog? [yes/no]: No
```

ステップ 4 スイッチ プロンプトで、特権 EXEC モードを開始します。

```
Switch> enable
Switch#
```

ステップ 5 スタートアップ コンフィギュレーションを実行コンフィギュレーションにコピーします。

```
Switch# copy startup-config running-config Destination filename [running-config]?
```

確認を求めるプロンプトに、Return を押して応答します。これで、コンフィギュレーション ファイルがリロードされ、パスワードを変更できます。

ステップ 6 グローバルコンフィギュレーションモードを開始して、**イネーブル**パスワードを変更します。

```
Switch# configure terminal
Switch(config)#
```

ステップ 7 実行コンフィギュレーションをスタートアップ コンフィギュレーション ファイルに書き込みます。

```
Switch(config)# copy running-config startup-config
```

ステップ 8 手動ブート モードがイネーブルになっていることを確認します。

```
Switch# show boot

 BOOT variable = flash:packages.conf;
 Manual Boot = yes
```

```
Enable Break = yes
```

ステップ 9 デバイスをリロードします。

```
Switch# reload
```

ステップ 10 SWITCH_IGNORE_STARTUP_CFG パラメータを 0 に設定します。

```
Switch(config)# no system ignore startupconfig switch all
Switch(config)# end
Switch# write memory
```

ステップ 11 フラッシュからのデバイス *packages.conf* を起動します。

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

ステップ 12 デバイス のブート後に、デバイス で手動ブートをディセーブルにします。

```
Switch(config)# no boot manual
```

パスワード回復がディセーブルになっている場合の手順

パスワード回復メカニズムがディセーブルの場合、次のメッセージが表示されます。

```
The password-recovery mechanism has been triggered, but
is currently disabled.  Access to the boot loader prompt
through the password-recovery mechanism is disallowed at
this point.  However, if you agree to let the system be
reset back to the default system configuration, access
to the boot loader prompt can still be allowed.

Would you like to reset the system back to the default configuration (y/n)?
```

⚠

注意 デバイスをデフォルト設定に戻すと、既存の設定がすべて失われます。システム管理者に問い合わせて、バックアップ デバイスと VLAN（仮想 LAN）コンフィギュレーション ファイルがあるかどうかを確認してください。

- **n**（no）を入力すると、Mode ボタンを押さなかった場合と同様に、通常のブート プロセスが継続されます。ブートローダ プロンプトにはアクセスできません。したがって、新しいパスワードを入力できません。次のメッセージが表示されます。

  ```
  Press Enter to continue........
  ```

- **y**（yes）を入力すると、フラッシュ メモリ内のコンフィギュレーション ファイルおよび VLAN データベース ファイルが削除されます。デフォルト設定がロードされるときに、パスワードをリセットできます。

手順

ステップ 1 パスワード回復手順の継続を選択すると、既存の設定が失われます。

```
Would you like to reset the system back to the default configuration (y/n)? Y
```

ステップ 2 フラッシュ メモリの内容を表示します。

```
Device: dir flash:
```

デバイスのファイル システムが表示されます。

```
Directory of flash:/
.
.
.i'
15494  drwx        4096   Jan 1 2000 00:20:20 +00:00  kirch
15508  -rw-   258065648   Sep 4 2013 14:19:03 +00:00
cat9k_caa-universalk9.SSA.03.12.02.EZP.150-12.02.EZP.150-12.02.EZP.bin
162196684
```

ステップ 3 システムを起動します。

```
Device: boot
```

セットアップ プログラムを起動するように求められます。パスワード回復手順を継続するには、プロンプトに **N** を入力します。

```
Continue with the configuration dialog? [yes/no]: N
```

ステップ 4 デバイス プロンプトで、特権 EXEC モードを開始します。

```
Device> enable
```

ステップ 5 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。

```
Device# configure terminal
```

ステップ 6 パスワードを変更します。

```
Device(config)# enable secret password
```

シークレット パスワードは 1 ～ 25 文字の英数字です。数字で始めることができます。大文字と小文字が区別され、スペースを使用できますが、先行スペースは無視されます。

ステップ 7 特権 EXEC モードに戻ります。

```
Device(config)# exit
Device#
```

ステップ 8 実行コンフィギュレーションをスタートアップ コンフィギュレーション ファイルに書き込みます。

```
Device# copy running-config startup-config
```

新しいパスワードがスタートアップ コンフィギュレーションに組み込まれました。

ステップ 9 ここでデバイスを再設定する必要があります。システム管理者によって、バックアップデバイスと VLAN コンフィギュレーション ファイルが使用可能に設定されている場合は、これらを使用します。

自動ネゴシエーションの不一致の防止

IEEE 802.3ab 自動ネゴシエーション プロトコルは速度（10 Mbps、100 Mbps、および SFP モジュール ポート以外の 1000 Mbps）およびデュプレックス（半二重または全二重）に関するデバイスの設定を管理します。このプロトコルは設定を適切に調整しないことがあり、その場合はパフォーマンスが低下します。不一致は次の条件で発生します。

- 手動で設定した速度またはデュプレックスのパラメータが、接続ポート上で手動で設定された速度またはデュプレックスのパラメータと異なっている場合。
- ポートを自動ネゴシエーションに設定したが、接続先ポートは自動ネゴシエーションを使用しない全二重に設定されている場合。

デバイスのパフォーマンスを最大限に引き出してリンクを確保するには、次のいずれかの注意事項に従って、デュプレックスおよび速度の設定を変更してください。

- 速度とデュプレックスの両方について、両方のポートで自動ネゴシエーションを実行させます。
- 接続の両側でポートの速度とデュプレックスのパラメータを手動で設定します。

（注） 接続先装置が自動ネゴシエーションを実行しない場合は、2 つのポートのデュプレックス設定を一致させます。速度パラメータは、接続先のポートが自動ネゴシエーションを実行しない場合でも自動調整が可能です。

SFP モジュールのセキュリティと識別に関するトラブルシューティング

シスコの Small Form-Factor Pluggable（SFP）モジュールは、モジュールのシリアル番号、ベンダー名とベンダー ID、一意のセキュリティ コード、および巡回冗長検査（CRC）が格納されたシリアル EEPROM（電気的に消去可能でプログラミング可能な ROM）を備えています。デバイスに SFP モジュールを装着すると、デバイス ソフトウェアは、EEPROM を読み取ってシリアル番号、ベンダー名、およびベンダー ID を確認し、セキュリティ コードおよび CRC を再計算します。シリアル番号、ベンダー名、ベンダー ID、セキュリティ コード、または CRC が無効な場合、ソフトウェアは、セキュリティ エラー メッセージを生成し、インターフェイスを errdisable ステートにします。

（注） セキュリティ エラー メッセージは、GBIC_SECURITY 機能を参照します。デバイスは、SFP モジュールをサポートしていますが、GBIC（ギガビット インターフェイス コンバータ）モジュールはサポートしていません。エラーメッセージテキストは、GBICインターフェイスおよびモジュールを参照しますが、セキュリティ メッセージは、実際は SFP モジュールおよびモジュール インターフェイスを参照します。

他社の SFP モジュールを使用している場合、デバイスから SFP モジュールを取り外し、シスコのモジュールに交換します。シスコの SFP モジュールを装着したら、**errdisable recovery cause gbic-invalid** グローバル コンフィギュレーション コマンドを使用してポート ステータスを確認し、error-disabled ステートから回復する時間間隔を入力します。この時間間隔が経過すると、デバイスは error-disabled ステートからインターフェイスを復帰させ、操作を再試行します。**errdisable recovery** コマンドの詳細については、このリリースに対応するコマンド リファレンスを参照してください。

モジュールがシスコ製SFPモジュールとして識別されたにもかかわらず、システムがベンダーデータ情報を読み取ってその情報が正確かどうかを確認できないと、SFP モジュール エラー メッセージが生成されます。この場合、SFPモジュールを取り外して再び装着してください。それでも障害が発生する場合は、SFP モジュールが不良品である可能性があります。

ping の実行

別の IP サブネットワーク内のホストに ping を実行する場合は、ネットワークへのスタティック ルートを定義するか、またはこれらのサブネット間でルーティングされるようにIPルーティングを設定する必要があります。

IP ルーティングは、デフォルトではすべてのデバイスでディセーブルになります。

（注） **ping** コマンドでは、他のプロトコル キーワードも使用可能ですが、このリリースではサポートされていません。

このコマンドは、デバイスからネットワーク上の他のデバイスに ping を実行する目的で使用します。

コマンド	目的
ping ip *host* | *address* `Device# ping 172.20.52.3`	IP またはホスト名やネットワーク アドレスを指定してリモート ホストに ping を実行します。

温度のモニタリング

デバイスは温度条件をモニタし、温度情報を使用してファンを制御します。

温度の値、状態、しきい値を表示するには、**show env temperature status** 特権 EXEC コマンドを使用します。温度の値は、デバイス内の温度であり、外部の温度ではありません。**system env temperature threshold yellow** *value* グローバル コンフィギュレーション コマンドを使用してイエローのしきい値レベル（摂氏）だけを設定し、イエローのしきい値およびレッドのしきい値の差を設定できます。グリーンまたはレッドのしきい値は設定できません。詳細については、このリリースのコマンド リファレンスを参照してください。

物理パスのモニタリング

次のいずれかの特権EXECコマンドを使用して、パケットが通過する、送信元デバイスから宛先デバイスへの物理パスをモニタできます。

表 *16 :* 物理パスのモニタリング

コマンド	目的
tracetroute mac [**interface** *interface-id*] {*source-mac-address*} [**interface** *interface-id*] {*destination-mac-address*} [**vlan** *vlan-id*] [**detail**]	指定の送信元MACアドレスから、指定の宛先MACアドレスまでをパケットが通過するレイヤ 2 パスを表示します。
tracetroute mac ip {*source-ip-address* | *source-hostname*}{*destination-ip-address* | *destination-hostname*} [**detail**]	指定の送信元 IP アドレスまたはホスト名から、指定の宛先 IP アドレスまたはホスト名を通過するパケットのレイヤ 2 パスを表示します。

IP traceroute の実行

（注） **traceroute** 特権 EXEC コマンドでは、他のプロトコル キーワードも使用可能ですが、このリリースではサポートされていません。

コマンド	目的
traceroute ip ホスト `Device# traceroute ip 192.51.100.1`	ネットワーク上でパケットが通過するパスを追跡します。

TDR の実行および結果の表示

TDR を実行する場合、**test cable-diagnostics tdr interface** *interface-id* 特権 EXEC コマンドを入力します。

TDR の結果を表示するには、**show cable-diagnostics tdr interface** *interface-id* 特権 EXEC コマンドを実行します。

デバッグおよびエラー メッセージ出力のリダイレクト

ネットワーク サーバはデフォルトで、**debug** コマンドおよびシステム エラー メッセージの出力をコンソールに送信します。このデフォルトの設定を使用する場合は、コンソールポートまたはイーサネット管理ポートに接続する代わりに、仮想端末接続によってデバッグ出力をモニタできます。

指定できる宛先として、コンソール、仮想端末、内部バッファ、およびsyslogサーバを実行している UNIX ホストがあります。Syslog フォーマットは、4.3 BSD UNIX およびそのバリエーションと互換性があります。

（注） デバッグの出力先がシステムのオーバーヘッドに影響を与えることがないように注意してください。メッセージをコンソールに記録すると、非常に高いオーバーヘッドが発生します。仮想端末にメッセージを記録すると、発生するオーバーヘッドは低くなります。Syslog サーバでメッセージロギングを行うと、オーバーヘッドはさらに小さくなり、内部バッファであれば最小限ですみます。

システム メッセージのロギングに関する詳細については、「システム メッセージ ロギングの設定」を参照してください。

show platform forward コマンドの使用

show platform forward 特権 EXEC コマンドの出力から、インターフェイスに入るパケットがシステムを介して送信された場合、転送結果に関して有意義な情報がいくつか得られます。パケットに関して入力されたパラメータに応じて、参照テーブル結果、転送宛先の計算に使用されるポート マップ、ビットマップ、および出力側の情報が表示されます。

このコマンドで出力される情報のほとんどは、主に、デバイスの用途別集積回路（ASIC）に関する詳細情報を使用するテクニカルサポート担当者に役立つものです。ただし、パケット転送情報はトラブルシューティングにも役立ちます。

show debug コマンドの使用方法

show debug コマンドは、特権 EXEC モードで入力します。このコマンドは、スイッチで使用可能なすべてのデバッグ オプションを表示します。

すべての条件付きデバッグ オプションを表示するには、コマンド **show debug condition** を実行します。コマンドは、条件 ID *<1-1000>*または *all* 条件を選択することで一覧表示できます。

デバッグを無効にするには、**no debug all** コマンドを使用します。

⚠

注意 デバッグ出力は CPU プロセスで高プライオリティが割り当てられているため、デバッグ出力を行うとシステムが使用できなくなることがあります。したがって、**debug** コマンドを使用するのは、特定の問題のトラブルシューティング時、またはシスコのテクニカルサポート担当者とともにトラブルシューティングを行う場合に限定してください。さらに、**debug** コマンドは、ネットワーク トラフィックが少なく、ユーザも少ないときに使用することを推奨します。デバッグをこのような時間帯に行うと、**debug** コマンド処理のオーバーヘッドの増加によりシステムの使用に影響が及ぶ可能性が少なくなります。

OBFL の設定

⚠

注意 OBFLはディセーブルにせず、フラッシュ メモリに保存されたデータは削除しないことを推奨します。

- OBFL をイネーブルにするには、**hw-switch switch** [*switch-number*] **logging onboard** [**message level** *level*] グローバル コンフィギュレーション コマンドを使用します。スイッチの場合、*switch-number* に指定できる範囲は 1 ～ 9 です。スイッチが生成してフラッシュ メモリに保存するハードウェア関連のメッセージの重大度を指定するには、**message level** *level* パラメータを使用します。
- OBFLデータをローカルネットワークまたは特定のファイル システムにコピーするには、**copy onboard switch** *switch-number* **url** *url-destination* 特権 EXEC コマンドを使用します。
- OBFL をディセーブルにするには、**no hw-switch switch** [*switch-number*] **logging onboard** [**message level**] グローバル コンフィギュレーション コマンドを使用します。
- フラッシュ メモリ内の稼働時間と CLI コマンド情報以外のすべての OBFL データをクリアするには、**clear onboard switch** *switch-number* 特権 EXEC コマンドを使用します。
- アクティブ スイッチのメンバ スイッチの OBFL をイネーブルまたはディセーブルにできます。

ここで説明した各コマンドの詳細については、このリリースのコマンドリファレンスを参照してください。

ソフトウェア設定のトラブルシューティングの確認

OBFL 情報の表示

表 *17 : OBFL* 情報を表示するためのコマンド

コマンド	目的
show onboard switch *switch-number* **clilog** `Device# show onboard switch 1 clilog`	スタンドアロン スイッチまたは指定されたスタック メンバで入力された OBFL CLI コマンドを表示します。
show onboard switch *switch-number* **environment** `Device# show onboard switch 1 environment`	スタンドアロン スイッチまたは指定されたスタック メンバおよび接続されているすべての FRU デバイスの UDI 情報、PID、VID、およびシリアル番号を表示します。
show onboard switch *switch-number* **message** `Device# show onboard switch 1 message`	スタンドアロン スイッチまたは指定されたスタック メンバによって生成されたハードウェア関連のメッセージを表示します。
show onboard switch *switch-number* **counter** `Device# show onboard switch 1 counter`	スタンドアロン スイッチまたは指定したスタック メンバのカウンタ情報を表示します。
show onboard switch *switch-number* **temperature** `Device# show onboard switch 1 temperature`	スタンドアロン スイッチまたは指定されたスイッチ スタック メンバの温度を表示します。
show onboard switch *switch-number* **uptime** `Device# show onboard switch 1 uptime`	スタンドアロン スイッチまたは指定されたスタック メンバが起動した時刻、スタンドアロン スイッチまたは指定されたスタック メンバが再起動された理由、およびスタンドアロン スイッチまたは指定されたスタック メンバが最後に再起動されて以来の稼働時間を表示します。
show onboard switch *switch-number* **voltage** `Device# show onboard switch 1 voltage`	スタンドアロン スイッチまたは指定されたスタック メンバのシステム電圧を表示します。

コマンド	目的
show onboard switch *switch-number* **status** `Device# show onboard switch 1 status`	スタンドアロン スイッチまたは指定されたスタック メンバの状態を表示します。

例：高い CPU 使用率に関する問題と原因の確認

CPU 使用率が高いことが問題となっているかどうか判別するには、**show processes cpu sorted** 特権 EXEC コマンドを入力します。出力例の 1 行目にある下線が付いた部分に注目してください。

```
Device# show processes cpu sorted
CPU utilization for five seconds: 8%/0%; one minute: 7%; five minutes: 8%
PID Runtime(ms) Invoked uSecs 5Sec 1Min 5Min TTY Process
309 42289103 752750 56180 1.75% 1.20% 1.22% 0 RIP Timers
140 8820183 4942081 1784 0.63% 0.37% 0.30% 0 HRPC qos request
100 3427318 16150534 212 0.47% 0.14% 0.11% 0 HRPC pm-counters
192 3093252 14081112 219 0.31% 0.14% 0.11% 0 Spanning Tree
143 8 37 216 0.15% 0.01% 0.00% 0 Exec
...
<output truncated>
```

この例は、正常な CPU 使用率を示しています。この出力によると、最後の 5 秒間の使用率が 8%/0% となっていますが、この意味は次のとおりです。

- Cisco IOS の処理時間と割り込みの処理にかかった時間を合わせた CPU の合計の使用率は全体の 8%
- 割り込みの処理にかかった時間は全体の 0%

表 *18 : CPU* 使用率に関する問題のトラブルシューティング

問題のタイプ	原因	修正処置
割り込みのパーセント値が合計の CPU 使用率の値とほぼ同程度に高い	CPU がネットワークから受信するパケット数が多すぎる。	ネットワーク パケットのソースを判別する。データの流れを遮断するか、スイッチの設定を変更します。「Analyzing Network Traffic（ネットワーク トラフィックの解析）」の項を参照してください。
割り込みの所要時間は最小限であったにもかかわらず CPU の合計使用率が 50% を超える	CPU 時間を過度に消費する Cisco IOS 処理が 1 つ以上存在する。これは通常、処理をアクティブ化するイベントによって始動されます。	異常なイベントを特定して根本的な原因を解消する。「Debugging Active Processes（アクティブなプロセスのデバッグ）」のセクションを参照してください。

ソフトウェア設定のトラブルシューティングのシナリオ

Power over Ethernet（PoE）に関するトラブルシューティングのシナリオ

表 *19 : Power over Ethernet* に関するトラブルシューティングのシナリオ

症状または問題	考えられる原因と解決法
PoE がないポートは 1 つに限りません。 1 つのスイッチ ポートに限り問題が発生する。このポートでは PoE 装置と PoE 非対応の装置のいずれも動作しないが、他のポートでは動作します。	

症状または問題	考えられる原因と解決法
	この受電デバイスが他のPoEポートで動作するかを確認する。 **show run**、または **show interface status** ユーザ EXEC コマンドを使用して、ポートがシャットダウンしていないか、または error-disabled になっていないかを確認します。 （注） ほとんどのスイッチはポートがシャットダウンしているときはポートの電力供給をオフにします。これは、IEEE 仕様でこれがオプションに指定されている場合も同様です。 **power inline never** がそのインターフェイスまたはポートで設定されていないことを確認します。 受電デバイスからスイッチポートまでのイーサネットケーブルの動作が正常であることを確認します。具体的には、既知の正常なPoE非対応のイーサネット装置とイーサネットケーブルを接続して、受電デバイスがリンクを確立し他のホストとトラフィックを交換することを確認します。 （注） シスコ受電デバイスは、ストレート ケーブルでのみ動作し、クロスケーブルでは動作しません。 スイッチのフロントパネルから受電デバイスまでのケーブル長の合計が 100 メートル以下であることを確認します。 スイッチ ポートからイーサネット ケーブルを外します。短いイーサネットケーブルを使用して、既知の正常なイーサネット装置を、スイッチのフロント パネルの（パッチパネルではない）このポートに直接接続します。これによってイーサネット リンクが確立され他のホストとトラフィックを交換できることを確認します。あるいは、ポートのVLAN SVI で ping を実行してください。次に、受電デバイスをこのポートに接続し、電源がオンになることを確認します。 パッチ コードをスイッチ ポートに接続しても受電デバイスの電源がオンにならない場合、接続する受電デバイスの合計数とスイッチの電力バジェット（使用可能な PoE）とを比較してください。**show inline power** コマンドを使用して、利用可能な電源の量を確認します。

症状または問題	考えられる原因と解決法
すべてのポートまたは1つのポート グループで PoE が機能しない。 すべてのスイッチ ポートで問題が発生する。電力が供給されていないイーサネット装置がどのポートでもイーサネット リンクを確立できず、PoE装置の電源がオンになりません。	

症状または問題	考えられる原因と解決法
	電力に関するアラームが継続的に発生する、断続的に発生する、または再発する場合は、可能であれば電源モジュールを交換します（現場交換可能ユニットです）。そうでない場合はスイッチを交換してください。 連続する複数のポートで問題があるものの、すべてのポートで問題が発生するわけではない場合、電源の故障ではないと考えられ、スイッチのPoEレギュレータに関連した異常の可能性があります。 PoEの状況やステータスの変更について過去に報告されているアラームまたはシステムメッセージを確認するには、**show log** 特権EXECコマンドを使用します。 アラームがない場合は、**show interface status** コマンドを使用して、ポートがシャットダウンしていないかerrdisableになっていないかを確認します。ポートがerror-disabledの場合、**shut** および **no shut** インターフェイス コンフィギュレーション コマンドを使用してポートを再びイネーブルにします。 特権EXECコマンドの **show env power** および **show power inline** を使用して、PoEのステータスおよび電力バジェット（使用可能なPoE）を調べます。 実行コンフィギュレーションを調べて **power inline never** がこのポートに設定されていないことを確認します。 受電していないイーサネット装置をスイッチポートに直接接続します。接続には短いパッチ コードだけを使用します。既存の配線ケーブルは使用しないでください。**shut** および **no shut** インターフェイス コンフィギュレーション コマンドを入力し、イーサネットリンクが確立されていることを確認します。正しく接続している場合、短いパッチコードを使用して受電デバイスをこのポートに接続し、電源がオンになることを確認します。装置の電源がオンになったら、すべての中間パッチパネルが正しく接続されているか確認してください。 1本を除くすべてのイーサネットケーブルをスイッチポートから抜きます。短いパッチ コードを使用して、1つのPoEポートにだけ受電デバイスを接続します。スイッチポートからの受電に比較して、受電デバイスが多くの電力を必要としないことを確認してください。 **show power inline** 特権EXECコマンドを使用して、ポートがシャットダウンしていない場合に、受電デバイスに電力が供給されることを確認します。あるいは、受電デバイス

症状または問題	考えられる原因と解決法
	を観察して電源がオンになることを確認してください。 1 台の受電デバイスだけがスイッチに接続しているときに電力が供給される場合、残りのポートで **shut** および **no shut** インターフェイス コンフィギュレーション コマンドを入力してから、イーサネット ケーブルをスイッチの PoE ポートに 1 本ずつ再び接続してください。**show interface status** および **show power inline** 特権 EXEC コマンドを使用して、インライン電源の統計情報およびポートの状態をモニタします。 すべてのポートで、まだ PoE が機能しない場合は、電源装置の PoE セクションでヒューズを開くことができる場合があります。この場合、アラームが生成されるのが一般的です。過去にシステム メッセージでアラームが報告されていないか、ログをもう一度チェックしてください。
シスコ先行標準受電デバイスは、切断またはリセットされます。 正常に動作した後で、Cisco phone またはワイヤレス アクセス ポイントが断続的にリロードしたり、PoE から切断されたりします。	スイッチから受電デバイスまでのすべての電気系統を確認してください。信頼性の低い接続は、電力供給の中断や受電デバイスの機能が不安定になる原因となり、受電デバイスの断続的な切断やリロードなどが発生します。 スイッチ ポートから受電デバイスまでのケーブル長が 100 メートル以下であることを確認してください。 スイッチが配置されている場所で電気環境にどのような変化があるか、切断時に、受電デバイスに何が起きるかについて注意してください。 切断と同時にエラー メッセージが表示されたか注意します。**show log** 特権 EXEC コマンドを使用してエラー メッセージを確認します。 リロードの発生直前に IP Phone から Call Manager へのアクセスが失われていないか確認してください（PoE の障害ではなくネットワークに問題が発生している場合があります）。 受電デバイスを PoE 非対応の装置に交換し、装置が正しく動作することを確認します。PoE 非対応の装置にリンク障害または高いエラー率がある場合、スイッチ ポートと受電デバイスを接続する信頼性の低いケーブル接続が問題の可能性があります。

症状または問題	考えられる原因と解決法
IEEE 802.3af 準拠または IEEE 802.3at 準拠の受電装置は、Cisco PoE スイッチでは機能しません。 シスコ PoE スイッチに接続するシスコ以外の受電デバイスに電源が供給されないか、電源投入後すぐに電源が切れます。PoE 非対応装置は正常に動作します。	**show power inline** コマンドを使用して、受電デバイスの接続前後に、スイッチの電力バジェット（使用可能な PoE）が使い果たされていないか確認してください。受電デバイスを接続する前に、このタイプの装置に十分な電力が使用可能であることを確認します。 **show interface status** コマンドを使用して、接続されている受電デバイスをスイッチが検出することを確認します。 **show log** コマンドを使用して、ポートの過電流状態を報告したシステム メッセージがないか確認します。症状を正確に特定してください。最初に電力が受電デバイスに供給され、その後、切断される状態ですか。その場合は、問題は最初のサージ電流（突入電流）が原因で、ポートの電流上限しきい値が超過した可能性があります。

ソフトウェアのトラブルシューティングの設定例

例：IP ホストの ping

次に、IP ホストに ping を実行する例を示します。

```
Device# ping 172.20.52.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echoes to 172.20.52.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Device#
```

表 *20 : Ping* の出力表示文字

文字	説明
!	感嘆符 1 個につき 1 回の応答を受信したことを示します。
.	ピリオド 1 個につき応答待ちの間にネットワーク サーバのタイムアウトが 1 回発生したことを示します。
U	宛先到達不能エラー PDU を受信したことを示します。
C	輻輳に遭遇したパケットを受信したことを示します。
I	ユーザによりテストが中断されたことを示します。

文字	説明
?	パケット タイプが不明です。
&	パケットの存続時間を超過したことを示します。

ping セッションを終了するには、エスケープ シーケンス（デフォルトでは Ctrl+^ X）を入力してください。Ctrl キー、Shift キー、および 6 キーを同時に押してから放し、その後 X キーを押します。

例：IP ホストに対する traceroute の実行

次に、IP ホストに **traceroute** を実行する例を示します。

```
Device# traceroute ip 192.0.2.10

Type escape sequence to abort.
Tracing the route to 192.0.2.10

  1 192.0.2.1 0 msec 0 msec 4 msec
  2 192.0.2.203 12 msec 8 msec 0 msec
  3 192.0.2.100 4 msec 0 msec 0 msec
  4 192.0.2.10 0 msec 4 msec 0 msec
```

ディスプレイには、送信される 3 つのプローブごとに、ホップ カウント、ルータの IP アドレス、およびラウンドトリップ タイム（ミリ秒単位）が表示されます。

表 *21 : traceroute* の出力表示文字

文字	説明
*	プローブがタイムアウトになりました。
?	パケット タイプが不明です。
A	管理上、到達不能です。通常、この出力は、アクセス リストがトラフィックをブロックしていることを表しています。
H	ホストが到達不能です。
N	ネットワークが到達不能です。
P	プロトコルが到達不能です。
Q	発信元。
U	ポートが到達不能です。

実行中の追跡を終了するには、エスケープ シーケンス（デフォルトでは Ctrl+^ X）を入力してください。Ctrl キー、Shift キー、および 6 キーを同時に押してから放し、その後 X キーを押します。

ソフトウェア設定のトラブルシューティングの機能履歴と情報

リリース	変更箇所
Cisco IOS XE Everest 16.5.1a	この機能が導入されました。