Cisco IOS XE Bengaluru 17.6.x（Catalyst 9300 スイッチ）システム管理 コンフィギュレーション ガイド

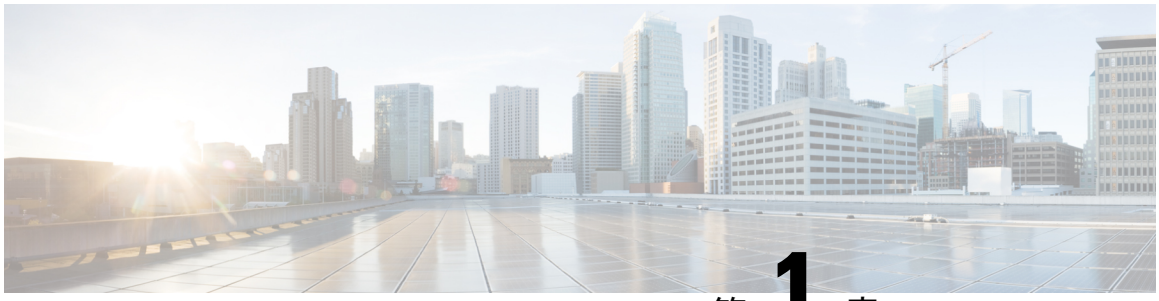
初版：2021 年 7 月 31 日

最終更新：2021 年 11 月 17 日

シスコシステムズ合同会社
〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー
http://www.cisco.com/jp
お問い合わせ先：シスコ コンタクトセンター
0120-092-255 （フリーコール、携帯・PHS含む）
電話受付時間：平日 10:00～12:00、13:00～17:00
http://www.cisco.com/jp/go/contactcenter/

【注意】シスコ製品をご使用になる前に、安全上の注意（ **www.cisco.com/jp/go/safety_warning/** ）をご確認ください。本書は、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。また、契約等の記述については、弊社販売パートナー、または、弊社担当者にご確認ください。

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。

- RTC
- 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。

- user **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 はユーザー 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 サーバー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、ネットワークにこの時刻を分配します。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 サーバーには、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 サーバーは、NTP を使用してストラタム 1 タイム サーバー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。

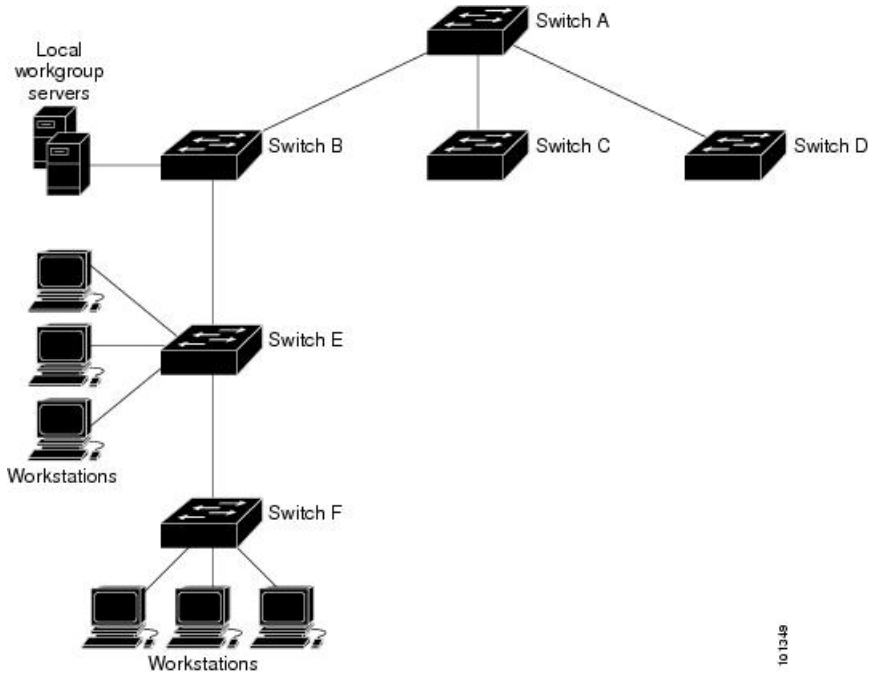
NTP が稼働するデバイス間の通信（アソシエーション）は、通常静的に設定されます。各デバイスには、アソシエーションを作成すべきすべてのデバイスの IP アドレスが与えられます。アソシエーションのペアとなるデバイス間で NTP メッセージを交換することによって、正確な時刻の維持が可能になります。ただし、LAN 環境では、代わりに IP ブロードキャスト メッセージを使用するように NTP を設定できます。各デバイスを、単にブロードキャスト メッセージを送受信するように設定すればよいので、この代替手段によって設定の複雑さが緩和されます。ただし、情報の流れは一方向に限られます。

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 サーバー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。A はプライマリ NTP、**デバイス** B、C、D が NTP サーバーモードに設定されている（デバイス A との間にサーバー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。デバイス E は、アップストリームデバイス（デバイス B）とダウンストリームデバイス（デバイス F）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 サーバーには、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 サーバーは、NTP を使用して

ストラタム 1 タイム サーバー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。

ポーリング ベースの *NTP* アソシエーション

NTP を実行しているネットワーキング デバイスは、時刻を基準時刻源と同期する際にさまざまなアソシエーション モードで動作するように設定できます。ネットワーキング デバイスは、2 つの方法でネットワーク上の時刻情報を取得できます。それらは、ホスト サービスのポーリングと NTP ブロードキャストのリスニングです。ここでは、ポーリング ベースのアソシエーション モードを中心に説明します。ブロードキャスト ベースの NTP アソシエーションの詳細については、「ブロードキャスト ベースの NTP アソシエーション」を参照してください。

最も一般的に使用される 2 つのポーリング ベースのアソシエーション モードは次のとおりです。

- クライアント モード
- 対称アクティブ モード

クライアント モードと対称アクティブ モードは、高レベルの時刻の精度と信頼性を提供するために NTP が必要になる場合に使用します。

クライアント モードで動作しているネットワーキング デバイスは、自身に割り当てられている時刻提供ホストをポーリングして現在の時刻を取得します。次に、ネットワーキング デバイスは、ポーリングされたすべてのタイム サーバーから、同期に使用するホストを選択します。この場合は、確立された関係がクライアント ホスト関係なので、ホストがローカル クライアント デバイスから送信された時刻情報をキャプチャしたり使用したりすることはありません。このモードが最も適しているのは、他のローカル クライアントにどのような形式の時刻同期も提供する必要のない、ファイル サーバーおよびワークステーションのクライアントです。ネットワーキング デバイスを同期させるタイム サーバーを個別に指定し、クライアント モードで動作するようにネットワーキング デバイスを設定するには、**ntp server** コマンドを使用します。

対称アクティブ モードで動作しているネットワーキング デバイスは、自身に割り当てられている時刻提供ホストをポーリングして現在の時刻を取得し、そのホストによるポーリングに応答します。これはピアツーピアの関係なので、ホストは、通信相手のローカル ネットワーキング デバイスの時刻関連情報も保持します。このモードは、さまざまなネットワーク パスを経由で多数の冗長サーバーが相互接続されている場合に使用します。インターネット上のほとんどの Stratum 1 および Stratum 2 サーバーは、この形式のネットワーク設定を採用しています。ネットワーキングデバイスを同期させる時刻提供ホストを個別に指定し、対称アクティブ モードで動作するようにネットワーキングデバイスを設定するには、**ntp peer** コマンドを使用します。

各ネットワーキング デバイスの設定モードを決定する際には、タイムキーピング デバイスとしてのそのデバイスの役割（サーバーかクライアントか）と、そのデバイスが Stratum 1 タイムキーピング サーバーにどれだけ近いかを主に考慮してください。

ネットワーキング デバイスは、クライアント モードでクライアントまたはホストとして動作する場合、または対称アクティブ モードでピアとして動作する場合にポーリングに関与します。通常、ポーリングによってメモリおよび CPU リソース（帯域幅など）に負荷が生じることはありませんが、システム上で進行または同時実行しているポーリングの数がきわめて多い場合には、システムのパフォーマンスに深刻な影響があったり、特定のネットワークのパフォーマンスが低下したりする可能性があります。過剰な数のポーリングがネットワーク上で進行することを防止するには、直接的なピアツーピア アソシエーションまたはクライアントからサーバーへのアソシエーションを制限する必要があります。代わりに、NTP ブロードキャストを使用して、ローカライズされたネットワーク内で時刻情報を伝播することを検討します。

ブロードキャスト ベースの *NTP* アソシエーション

ブロードキャストベースの NTP アソシエーションは、時刻の精度および信頼性要件が適度であり、ネットワークがローカライズされ、クライアント数が 20 を超える場合に使用します。また、帯域幅、システム メモリ、または CPU リソースが制限されているネットワークにおいても、ブロードキャストベースの NTP アソシエーションの使用をお勧めします。

ブロードキャスト クライアント モードで動作しているネットワーキング デバイスはポーリングに関与しません。代わりに、ブロードキャスト タイム サーバーによって転送される NTP ブロードキャストパケットをリッスンします。その結果、時刻情報の流れが一方向に限られるため、時刻の精度がわずかに低下する可能性があります。

ネットワークを通じて伝播される NTP ブロードキャストパケットをリッスンするようにネットワーキングデバイスを設定するには、**ntp broadcast client** コマンドを使用します。ブロードキャスト クライアント モードが動作するためには、ブロードキャスト サーバーとそのクライアントが同じサブネット上に存在する必要があります。**ntp broadcast** コマンドを使用して、特定のデバイスのインターフェイスで NTP ブロードキャスト ケットを送信するタイムサーバーを有効にする必要があります。

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。

（注） Message Direct 5（MD5）認証の設定は推奨しません。より強力な暗号化のためにサポートされている他の認証方式を使用できます。

NTP アクセス グループ

アクセス リストベースの制限スキームを使用すると、ネットワーク全体、ネットワーク内のサブネット、またはサブネット内のホストに対し、特定のアクセス権限を許可または拒否できます。NTP アクセスグループを定義するには、グローバル コンフィギュレーション モードで ntp access-group コマンドを使用します。

アクセス グループのオプションは、次の順序で制限の緩いものから厳しいものへとスキャンされます。

1. ipv4：IPv4 アクセスリストを設定します。
2. ipv6：IPv6 アクセスリストを設定します。
3. peer：時刻要求と NTP 制御クエリを許可し、システムがアクセスリストの基準を満たすアドレスを持つ別のシステムに同期することを許可します。
4. serve：時刻要求と NTP 制御クエリを許可しますが、システムがアクセスリストの基準を満たすアドレスを持つ別のシステムに同期することは許可しません。
5. serve-only：アクセスリストの条件を満たすアドレスを持つシステムからの時刻要求のみを許可します。
6. query-only：アクセスリストの基準を満たすアドレスを持つ別のシステムからの NTP 制御クエリのみを許可します。

送信元 IP アドレスが複数のアクセス タイプのアクセス リストに一致する場合は、最初のアクセス タイプのアクセスが認可されます。アクセス グループが指定されていない場合は、すべてのシステムへのアクセスがすべてのアクセス タイプに対して認可されます。アクセス グループが指定されている場合は、指定されたアクセス タイプに対してのみアクセスが認可されます。

NTP 制御クエリーの詳細については、RFC 1305（NTP バージョン 3）を参照してください。

信頼できる形式のアクセス コントロールが必要な場合は、暗号化された NTP 認証方式を使用する必要があります。IP アドレスに基づくアクセス リスト ベースの制約方式とは異なり、暗号化認証方式では、認証キーと認証プロセスを使用して、ローカル ネットワーク上の指定されたピアまたはサーバーによって送信された NTP 同期パケットが信頼できると見なされるかどうかを、一緒に伝送された時刻情報を受け入れる前に判断します。

認証プロセスは、NTP パケットが作成されるとすぐに開始されます。暗号チェックサム キーは、Message-Digest Algorithm 5（MD5）を使用して生成され、受信側クライアントに送信される NTP 同期パケットに埋め込まれます。パケットがクライアントによって受信されると、暗号チェックサム キーが復号され、信頼できるキーのリストに対してチェックされます。一致する認証キーがパケットに含まれる場合、受信側クライアントは、パケットに含まれるタイムス

タンプ情報を受け入れます。一致するオーセンティケータ キーが含まれていない NTP 同期パケットは無視されます。

（注） 信頼できるキーを多数設定する必要がある大規模なネットワークでは、信頼できるキーの範囲設定機能を使用して複数のキーを同時にイネーブルにすることができます。

NTP 認証で使用される暗号化および復号化プロセスでは、CPU に非常に大きな負荷がかかる場合があり、ネットワーク内で伝播される時刻の精度が大きく低下する可能性があることに注意してください。より包括的なアクセス コントロール モデルを使用できるネットワーク構成の場合は、アクセス リスト ベースのコントロール方式を使用することを検討してください。

NTP認証が適切に設定されると、ネットワーキングデバイスは、信頼できる時刻源と同期し、信頼できる時刻源だけに同期を提供します。

特定のインターフェイス上の NTP サービス

Network Time Protocol（NTP）サービスは、デフォルトではすべてのインターフェイスでディセーブルになっています。なんらかのNTPコマンドを入力すると、NTPがグローバルにイネーブルになります。特定のインターフェイスを通じて特定の NTP パケットを受信しないように設定するには、インターフェイス コンフィギュレーション モードで **ntp disable** コマンドを使用します。

NTP パケットの送信元 IP アドレス

システムがNTPパケットを送信すると、通常、送信元IPアドレスは、そのNTPパケットの送信元であるインターフェイスのアドレスに設定されます。IP 送信元アドレスの取得元のインターフェイスを設定するには、グローバル コンフィギュレーション モードで **ntp source** *interface* コマンドを使用します。

このインターフェイスは、すべての宛先に送信されるすべてのパケットの送信元アドレスに使用されます。特定のアソシエーションに送信元アドレスを使用する場合は、**ntp peer** コマンドまたは **ntp server** コマンドで **source** キーワードを使用します。

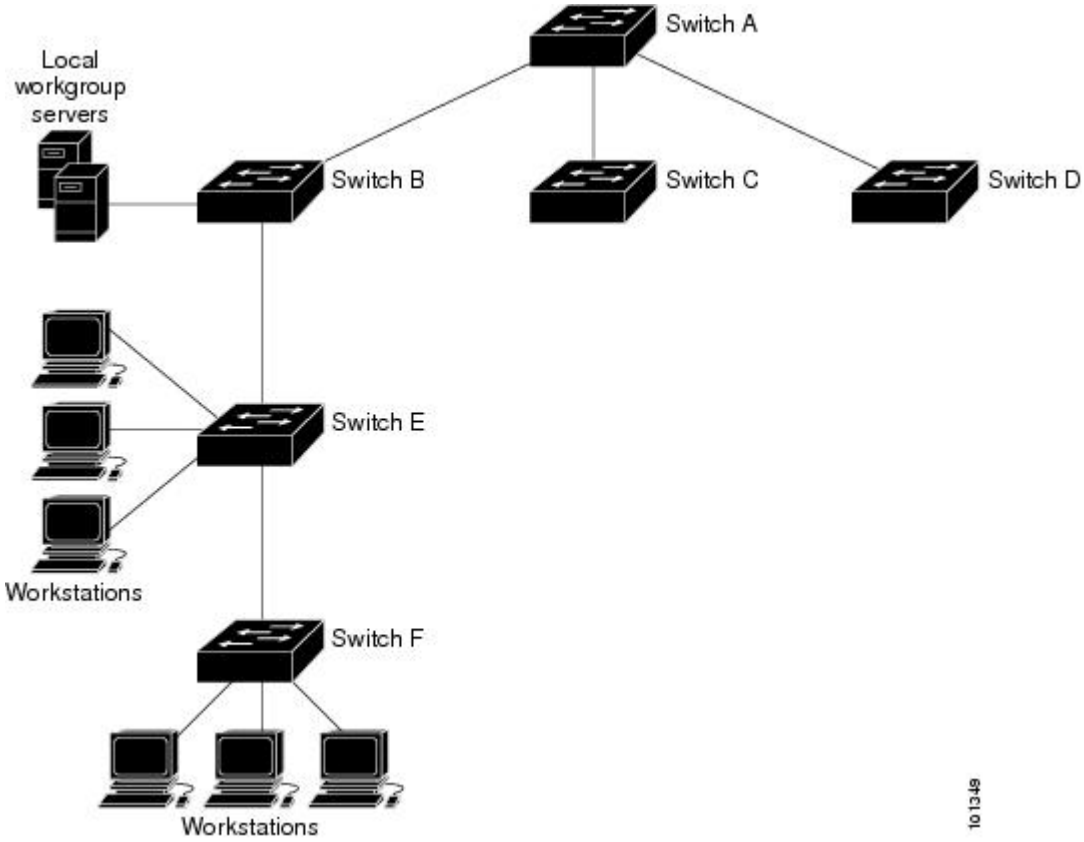
NTP の実装

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 サーバー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サーバーモードに設定されている（スイッチAとの間にサーバーアソシエーションが設定されている）場合のプライマリ NTP 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 コマンドを使用する場合や、関連する Telnet サポート操作時に、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 サーバーという概念が定義されています。ドメイン ネーム サーバーの役割は、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 サーバーを指定し、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	イネーブル
DNS デフォルト ドメイン名	未設定
DNS サーバー	ネーム サーバー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 ユーザー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
エージング タイム	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。

デバイスの管理方法

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。

（注） 手動でシステムクロックを設定している場合は、デバイスに障害が発生して別のスタックメンバがデバイスの役割を引き継ぐ前に、この設定を再設定する必要があります。

システム クロックの設定

ネットワーク上に、NTP サーバー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 モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	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。

	コマンドまたはアクション	目的
		• *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 モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル設定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：	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。

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 モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル設定モードを開始します。
ステップ 3	**clock summer-time** *zone* **date** *date month year hh:mm date month year hh:mm* [*offset*]] 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。

	コマンドまたはアクション	目的
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 PDT recurring 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。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。 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、夏時間のルールは米国のルールにデフォルト設定されます。 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。

	コマンドまたはアクション	目的
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。

NTP の設定

デバイスはハードウェアサポート クロックを備えておらず、外部 NTP ソースが利用できないときに、ピアが自身を同期化するための NTP プライマリクロックとして機能することはできません。デバイスは、カレンダーに対するハードウェアサポートも備えていません。そのため、グローバル コンフィギュレーション モードで **ntp update-calendar** コマンドと **ntp master** コマンドを使用することはできません。

NTP の設定情報については、次のセクションを参照してください。

NTP のデフォルト設定

NTP のデフォルト設定を示します。

表 *3: NTP* のデフォルト設定

機能	デフォルト設定
NTP 認証	ディセーブル認証キーは指定されていません。
NTP ピアまたはサーバー アソシエーション	未設定
NTP ブロードキャスト サービス	ディセーブル。どのインターフェイスも NTP ブロードキャストパケットを送受信しません。
NTP アクセス制限	アクセスコントロールは指定されていません。
NTP パケット送信元 IP アドレス	送信元アドレスは、発信インターフェイスによって設定されます。

NTPは、すべてのインターフェイスでデフォルトでイネーブルに設定されています。すべてのインターフェイスは、NTP パケットを受信します。

NTP 認証の設定

NTP 認証を設定するには、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 モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル設定モードを開始します。
ステップ 3	[**no**] **ntp authenticate** 例： `Device(config)# ntp authenticate`	NTP 認証をイネーブルにします。 NTP 認証を無効にするには、このコマンドの **no** 形式を使用します。
ステップ 4	[**no**] **ntp authentication-key** *number* {**md5** | **cmac-aes-128** | **hmac-sha1** | **hmac-sha2-256**} *value* 例： `Device(config)# ntp authentication-key 42 md5 aNiceKey`	認証キーを定義します。 • キーごとに、キー番号、タイプ、および値を 1 つずつ指定します。 • キーは次のいずれかのタイプになります。 • **md5**：MD5 アルゴリズムを使用した認証。 • **cmac-aes-128**：AES-128 アルゴリズムによる暗号ベースメッセージ承認コード（CMAC）を使用した認証。ダイジェストの長さは 128 ビットで、キーの長さは 16 バイトまたは 32 バイトです。 • **hmac-sha1**：SHA1ハッシュ関数を使用したハッシュベースメッセージ承認コード（HMAC）を使用した認証。ダイジェストの長さは 128 ビットで、キーの長さは 1 ー 32 バイトです。

	コマンドまたはアクション	目的
		• **hmac-sha2-256**：SHA2 ハッシュ関数を使用した HMAC を使用した認証。ダイジェストの長さは 256 ビットで、キーの長さは 1 ー 32 バイトです。 SNTP の認証キーを削除する場合は、このコマンドの **no** 形式を使用します。
ステップ 5	[**no**] **ntp trusted-key** *key-number* 例： `Device(config)# ntp trusted-key 42`	このデバイスと同期できるようにするために、ピア NTP デバイスが NTP パケットで提供する必要がある信頼できる認証キーを定義します。 信頼できる認証を無効にするには、このコマンドの **no** 形式を使用します。
ステップ 6	[**no**] **ntp server** *ip-address* **key** *key-id* [**prefer**] 例： `Device(config)# ntp server 172.16.22.44 key 42`	NTP タイムサーバーによってソフトウェア クロックが同期されるように設定します。 • *ip-address*：クロック同期を提供するタイムサーバーの IP アドレス。 • *key-id*：**ntp authentication-key** コマンドで定義された認証キー。 • **prefer**：このピアを、同期を提供する優先ピアにします。このキーワードは、ピア間のクロックホップを減らします。 サーバーアソシエーションを解除するには、このコマンドの **no** 形式を入力します。
ステップ 7	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

ポーリング ベースの NTP アソシエーションの設定

ポーリングベースの NTP アソシエーションを設定するには、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 モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル設定モードを開始します。
ステップ 3	[**no**] **ntp peer** *ip-address* [**version** *number*] [**key** *key-id*] [**source** *interface*] [**prefer**] 例： `Device(config)# ntp peer 172.16.22.44 version 2`	ピアを同期化するか、またはピアによって同期化されるように、デバイスのシステムクロックを設定します（ピアアソシエーション）。 • *ip-address*：クロック同期を提供する、またはクロック同期を提供されるピアの IP アドレス。 • *number*：NTP バージョン番号。範囲は、1 ～ 3 です。デフォルトでは、バージョン3が選択されています。 • *key-id*：**ntp authentication-key** コマンドで定義された認証キー。 • *interface*：IP の送信元アドレスを取得するインターフェイス。デフォルトでは、送信元 IP アドレスは発信インターフェイスから取得します。 • **prefer**：このピアを、同期を提供する優先ピアにします。このキーワードにより、ピア間の切り替えが減少します。 ピアアソシエーションを解除するには、このコマンドの **no** 形式を使用します。
ステップ 4	[**no**] **ntp server** *ip-address* [**version** *number*] [**key** *key-id*] [**source** *interface*] [**prefer**] 例：	タイムサーバーによって同期化されるように、デバイスのシステムクロックを設定します（サーバーアソシエーション）。

	コマンドまたはアクション	目的
	`Device(config)# ntp server 172.16.22.44 version 2`	• *ip-address*：クロック同期を提供するタイムサーバーの IP アドレス。 • *number*：NTP バージョン番号。範囲は、1 ～ 3 です。デフォルトでは、バージョン3が選択されています。 • *key-id*：**ntp authentication-key** コマンドで定義された認証キー。 • *interface*：IP の送信元アドレスを取得するインターフェイス。デフォルトでは、送信元 IP アドレスは発信インターフェイスから取得します。 • **prefer**：このピアを、同期を提供する優先ピアにします。このキーワードは、ピア間のクロックホップを減らします。 サーバーアソシエーションを解除するには、このコマンドの **no** 形式を入力します。
ステップ 5	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

ブロードキャスト ベースの NTP アソシエーションの設定

ブロードキャストベースの NTP アソシエーションを設定するには、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 モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例：	グローバル設定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	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet1/0/1`	インターフェイスを設定し、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 4	[**no**] **ntp broadcast** [**version** *number*] [**key** *key-id*] [*destination-address*] 例： `Device(config-if)# ntp broadcast version 2`	NTP ブロードキャスト パケットをピアに送信するインターフェイスをイネーブルにします。 • *number*：NTP バージョン番号。範囲は、1 ～ 3 です。デフォルトでは、バージョン3が使用されます。 • *key-id*：認証キー。 • *destination-address*：このスイッチに対してクロックを同期しているピアの IP アドレス。 インターフェイスでのNTPブロードキャストパケットの送信を無効にするには、このコマンドの **no** 形式を使用します。
ステップ 5	[**no**] **ntp broadcast client** 例： `Device(config-if)# ntp broadcast client`	インターフェイスが NTP ブロードキャスト パケットを受信できるようにします。 インターフェイスでのNTPブロードキャストパケットの受信を無効にするには、このコマンドの **no** 形式を使用します。
ステップ 6	**exit** 例： `Device(config-if)# exit`	特権 EXEC モードに戻ります。
ステップ 7	[**no**] **ntp broadcastdelay** *microseconds* 例： `Device(config)# ntp broadcastdelay 100`	（任意）デバイスと NTP ブロードキャストサーバー間のラウンドトリップ遅延の予測値を変更します。 デフォルトは 3000 マイクロ秒です。範囲は 1 ～ 999999 です。

	コマンドまたはアクション	目的
		インターフェイスでのNTPブロードキャストパケットの受信を無効にするには、このコマンドの **no** 形式を使用します。
ステップ 8	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

NTP アクセス制限の設定

以降で説明するように、2 つのレベルで NTP アクセスを制御できます。

アクセス グループの作成と基本 IP アクセス リストの割り当て

アクセスグループを作成して基本 IP アクセスリストを割り当てるには、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 モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル設定モードを開始します。
ステップ 3	[**no**] **ntp access-group** {**query-only** | **serve-only** | **serve** | **peer**} *access-list-number* 例： `Device(config)# ntp access-group peer 99`	アクセスグループを作成し、基本 IP アクセスリストを割り当てます。 • **query-only**：NTP 制御クエリ。 • **serve-only**：時間要求。 • **serve**：時刻要求と NTP 制御クエリは許可しますが、リモートデバイスに対するデバイスの同期化は許可しません。

	コマンドまたはアクション	目的
		• **peer**：時刻要求と NTP 制御クエリ、およびリモートデバイスに対するデバイスの同期化を許可します。 • *access-list-number*：IP アクセスリスト番号。指定できる範囲は 1 ～ 99 です。 スイッチ NTP サービスに対するアクセス制御を削除するには、このコマンドの **no** 形式を使用します。
ステップ 4	**access-list** *access-list-number* **permit** *source* [*source-wildcard*] 例： `Device(config)# access-list 99 permit 172.20.130.5`	アクセス リストを作成します。 • *access-list-number*：IP アクセスリスト番号。指定できる範囲は 1 ～ 99 です。 • **permit**：条件が一致した場合にアクセスを許可します。 • *source*：デバイスへのアクセスが許可されているデバイスの IP アドレス。 • *source-wildcard*：送信元アドレスに適用されるワイルドカードビット。 （注） アクセス リストを作成する際は、アクセス リストの末尾に暗黙の deny ステートメントがデフォルトで存在し、ACL の終わりに到達するまで一致が見つからなかったすべてのパケットに適用されることに注意してください。 SNTP の認証キーを削除する場合は、このコマンドの **no** 形式を使用します。
ステップ 5	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

特定のインターフェイス上の NTP サービスのディセーブル化

インターフェイスで NTP パケットの受信を無効にするには、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 モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル設定モードを開始します。
ステップ 3	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet1/0/1`	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	[**no**] **ntp disable** 例： `Device(config-if)# ntp disable`	インターフェイスで NTP パケットの受信をディセーブルにします。 インターフェイスで NTP パケットの受信を再度有効にするには、このコマンドの **no** 形式を使用します。
ステップ 5	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。

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 モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル設定モードを開始します。
ステップ 3	**hostname** *name* 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 モードで **ip domain name** コマンドによって設定される値です。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 モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル設定モードを開始します。
ステップ 3	**ip domain name** *name* 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）サーバーから行われている場合、BOOTP または DHCP サーバー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（この情報がサーバー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：	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 サーバーのアドレスを指定します。

	コマンドまたはアクション	目的
	`Device(config)# ip name-server 192.168.1.100 192.168.1.200 192.168.1.300`	最大 6 つのネーム サーバーを指定できます。各サーバー 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ーが、プライマリ サーバーです。デバイスは、プライマリサーバーへ DNS クエリを最初に送信します。そのクエリが失敗した場合は、バックアップ サーバー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されています。 ユーザー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、ユーザー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。
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 モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル設定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` `authorized users are allowed.` `For access, contact technical` `support.` `#`	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 モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル設定モードを開始します。
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：	（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
	`Device# copy running-config startup-config`	

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 モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル設定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。 指定できる範囲は 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。

	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。

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 モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル設定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 情報を送信します。 • **version**：サポートする SNMP バージョンを指定します。informs には

	コマンドまたはアクション	目的
		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** 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から、**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。 • **vrf** *vrf* インスタンス名：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 アドレス変更通知を NMS に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。

	コマンドまたはアクション	目的
		• （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： `Device(config-if)# snmp trap mac-notification change added`	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。 • MACアドレスがインターフェイスに**added**された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に**removed**された場合に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。

次の手順に従い、デバイスを設定し、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 モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル設定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 情報を送信します。 • **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** 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から、**snmp-server host** コマンドを使用することを推奨します。

	コマンドまたはアクション	目的
		• *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 notification mac-move`	MAC アドレス移動通知機能をイネーブルにします。
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 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

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 アドレス移動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 モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル設定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 情報を送信します。 • **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** 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から、**snmp-server host** コマンドを使用することを推奨します。

	コマンドまたはアクション	目的
		• *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 notification threshold`	MAC アドレスしきい値通知機能をイネーブルにします。
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ー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`	（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`	

VLAN の MAC アドレスラーニングのディセーブル化

VLAN で MAC アドレスラーニングを制御すると、MAC アドレスを学習できる VLAN を制御することで、利用可能な MAC アドレステーブルスペースを管理できます。MAC アドレスラーニングをディセーブルにする前に、ネットワークトポロジをよく理解しておいてください。VLAN で MAC アドレスラーニングをディセーブルにすると、ネットワークでフラッディングを引き起こす可能性があります。

VLAN で MAC アドレス ラーニングをディセーブルにするには、特権 EXEC モードで次の手順を実行します。

始める前に

VLAN の MAC アドレス ラーニングをディセーブルにする際は、次の注意事項に従ってください。

- スイッチ仮想インターフェイス（SVI）スイッチを設定済みの VLAN で MAC アドレス ラーニングをディセーブルにする場合は、十分注意してください。この場合、スイッチはレイヤ 2 ドメインにすべての IP パケットをフラッディングします。
- MAC アドレスラーニングは、2 から 4094 までの 1 つの VLAN ID（例：no mac address-table learning vlan 223）、または、ハイフンやカンマで区切られた一連の VLAN ID（例：no mac address-table learning vlan 1-10, 15）でディセーブルにできます。
- MAC アドレスラーニングのディセーブル化は、ポートを 2 つ含む VLAN だけで行うことを推奨します。3 つ以上のポートを含む VLAN で MAC アドレス ラーニングをディセーブルにした場合は、スイッチに着信するすべてのパケットは、その VLAN ドメインでフラッディングします。
- セキュア ポートを含む VLAN で MAC アドレス ラーニングをディセーブルにする場合、そのポートで MAC アドレス ラーニングはディセーブルになりません。

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 mac-address-table learning vlan**[*vlan-id* |,*vlan-id* | -*vlan-id*,] 例：	指定された 1 つまたは複数の VLAN で MAC アドレスラーニ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
	`Device(config)# no mac-address-table learning {vlan vlan-id [,vlan-id | -vlan-id]`	1 つの VLAN ID を指定、または VLAN ID の範囲をハイフンまたはカンマで区切って指定できます。有効な VLAN ID の範囲は 2 ～ 4094 です。内部 VLAN は指定できません。
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 mac-address-table learning vlan**[*vlan-id*] 例： `Device# show mac-address-table learning [vlan vlan-id]`	設定を確認します。 show mac-address-table learning [vlan vlan-id] 特権 EXEC コマンドを入力すると、すべての VLAN、または指定した VLAN の MAC アドレス ラーニングのステータスを表示できます。
ステップ 5	**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。
ステップ 6	**default mac address-table learning** 例： `Device# default mac address-table`	（任意）グローバル コンフィギュレーション モードで VLAN の MAC アドレスラーニングを再度イネーブルにします。

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 モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル設定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	**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 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 モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル設定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。
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。

デバイスのモニター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。
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。

コマンド	目的
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	マニュアル タイトル
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 9300* シリーズ スイッチ）

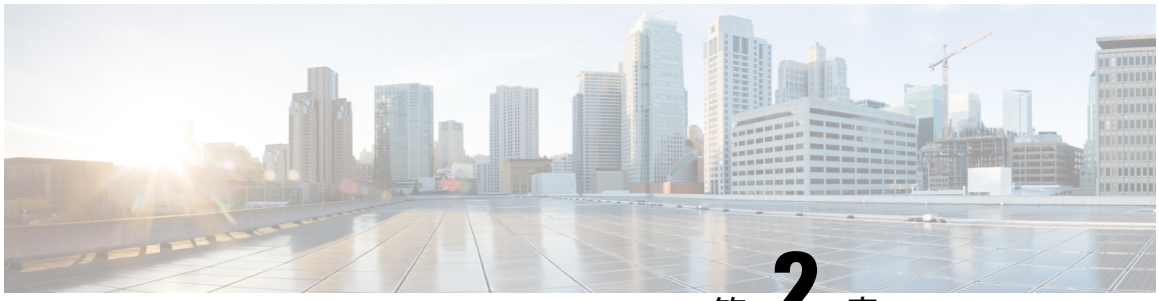
デバイス管理の機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

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

リリース	機能	機能情報
Cisco IOS XE Everest 16.5.1a	デバイス管理	デバイス管理では、システムの日時、システム名、ログインバナーを設定し、DNS を設定できます。

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



第 2 章

ブート整合性の可視性

ブート整合性の可視性について

ブート整合性の可視性によって、シスコのプラットフォーム ID とソフトウェアの整合性情報が可視化され、実用可能になります。プラットフォーム ID は、プラットフォームの製造元でインストールされた ID を提供します。ソフトウェアの整合性ではブート整合性の測定値が明らかになり、それを使用してプラットフォームが信頼できるコードを起動しているかどうかを評価できます。

ブートプロセス中に、ソフトウェアはブートローダーアクティビティの各ステージのチェックサムレコードを作成します。

このレコードを取得して、シスコ認定レコードと比較し、ソフトウェアイメージが正規かどうかを確認できます。チェックサム値が一致していない場合は、シスコによって認定されていない、または未承認パーティによって改ざんされているソフトウェアイメージを実行している可能性があります。

イメージ署名とブートアップ

シスコの構築したサーバーが Cisco IOS XE イメージを生成します。Cisco IOS XE イメージの場合、Abraxas イメージ署名システムを使用して、シスコの秘密 RSA キーでイメージに安全に署名できます。

Cisco IOS XE イメージを Catalyst 9000 シリーズ スイッチにコピーすると、シスコの ROMMON ブート ROM がシスコのリリースキーを使用してイメージを検証します。これらのキーは、

Abraxas サーバーに安全に保存されているシスコのリリース秘密キーに対応する公開キーです。リリース秘密キーは ROMMON に保存されます。

Catalyst 9000 シリーズ スイッチは、ブート整合性の可視性機能をサポートしています。ブート整合性の可視性は、ROMMON ソフトウェアが改ざんされていないことを確認するために、ROMMON ソフトウェアを検証するハードウェア トラスト アンカーとして機能します。

Cisco IOS XE イメージは、構築時にデジタル署名されます。バイナリイメージファイル全体に対して SHA-512 ハッシュが生成され、このハッシュがシスコの RSA 2048 ビット秘密キーで暗号化されます。ROMMON は、シスコの公開キーを使用して署名を検証します。このソフトウェアがシスコの構築したシステムによって生成されたものではない場合、署名の検証は失敗します。デバイスの ROMMON はイメージを拒否し、起動を停止します。署名の検証に成功すると、デバイスはイメージを Cisco IOS XE ランタイム環境で起動します。

ROMMON は、ブートアップ中に署名付き Cisco IOS XE イメージを検証する際、次の手順を実行します。

1. Cisco IOS XE イメージを CPU メモリにロードします。
2. Cisco IOS XE パッケージのヘッダーを調べます。
3. イメージに対して非セキュア整合性チェックを実行し、ディスクまたは TFTP で意図しないファイル破損が生じていないことを確認します。これは非セキュア SHA-1 ハッシュを使用して実行されます。
4. シスコの RSA 2048 ビット公開リリースキーを ROMMON ストレージからコピーし、シスコの RSA 2048 ビット公開リリースキーが改ざんされていないことを検証します。
5. パッケージのヘッダーからコード署名用署名（SHA-512 ハッシュ）を抽出し、シスコの RSA 2048 ビット公開キーを使用して検証します。
6. Cisco IOS XE パッケージの SHA-512 ハッシュを計算してコード署名の検証を実行し、コード署名用署名と比較します。これで署名付きパッケージの検証が実行されたことになります。
7. Cisco IOS XE パッケージのヘッダーを調べて、プラットフォームタイプと CPU アーキテクチャの互換性を検証します。
8. Cisco IOS XE パッケージから Cisco IOS XE ソフトウェアを抽出して起動します。

（注） 上記のプロセス中、手順 3 はイメージの非セキュアチェックであり、ディスクエラー、ファイル転送エラー、またはコピーエラーによる偶発的な破損に関してイメージを確認することを目的としています。これはイメージコード署名の一環ではありません。このチェックは、意図的なイメージの改ざんを検出するためのものではありません。

イメージコード署名の検証は、手順 4、5、および 6 で行われます。これは、2048 ビット RSA キーで暗号化された SHA-512 ハッシュを使用した、イメージのセキュアコード署名チェックです。このチェックは、意図的なイメージの改ざんを検出することを目的としています。

ソフトウェアイメージとハードウェアの確認

このタスクでは、スイッチの起動時に作成されたチェックサムレコードを取得する方法について説明します。特権 EXEC モードで次のコマンドを入力します。

（注） 次のコマンドを実行した後で、メッセージ **% Please Try After Few Seconds** が CLI に表示されることがあります。これはCLIの障害を示すものではありませんが、必要な出力を取得するために必要な基盤となるインフラストラクチャの設定を示します。数分間待機して、コマンドを再度試すことをお勧めします。

メッセージ **% Error retrieving SUDI certificate** および **% Error retrieving integrity data** は、実際の CLI 障害を示します。

手順

	コマンドまたはアクション	目的
ステップ 1	**show platform sudi certificate** ［**sign** ［**nonce** *nonce*］］ 例： `Device# show platform sudi certificate sign nonce 123`	特定の SUDI のチェックサム レコードを表示します。 • （オプション）**sign**：署名を示します • （オプション）**nonce**：ナンス値を入力します
ステップ 2	**show platform integrity** ［**sign** ［**nonce** *nonce*］］ 例： `Device# show platform integrity sign nonce 123`	ブート段階のチェックサム レコードを表示します。 • （オプション）**sign**：署名を示します • （オプション）**nonce**：ナンス値を入力します

プラットフォーム ID とソフトウェア整合性の確認

プラットフォーム ID の確認

次に、PEM形式でセキュアな固有デバイス識別子（SUDI）チェーンを表示する例を示します。SUDIにエンコードされるのは、個々のデバイスの製品IDとシリアル番号であり、何千ものデバイスからなるネットワーク上でデバイスを一意に識別できます。最初の証明書は Cisco Root CA 2048 で、2 番目はシスコの下位 CA（ACT2 SUDI CA）です。どちらの証明書も、

https://www.cisco.com/security/pki/ で公開されているものと一致しているかを確認できます。3 番目は SUDI 証明書です。

```
 Device# show platform sudi certificate sign nonce 123
-----BEGIN CERTIFICATE-----
MIIDQzCCAiugAwIBAgIQX/h7KCtU3I1CoxW1aMmt/zANBgkqhkiG9w0BAQUFADA1
MRYwFAYDVQQKEw1DaXNjbyBTeXN0ZW1zMRswGQYDVQQDExJDaXNjbyBSb290IENB
IDIwNDgwHhcNMDQwNTE0MjAxNzEyWhcNMjkwNTE0MjAyNTQyWjA1MRYwFAYDVQQK
Ew1DaXNjbyBTeXN0ZW1zMRswGQYDVQQDExJDaXNjbyBSb290IENBIDIwNDgwggEg
MA0GCSqGSIb3DQEBAQUAA4IBDQAwggEIAoIBAQCwmrmrp68Kd6ficba0ZmKUeIhH
xmJVhEAyv8CrLqUccda8bnuoqrpu0hWISEWdovyD0My5jOAmaHBKeN8hF570YQXJ
FcjPFto1YYmUQ6iEqDGYeJu5Tm8sUxJszR2tKyS7McQr/4NEb7Y9JHcJ6r8qqB9q
VvYgDxFUl4F1pyXOWWqCZe+36ufijXWLbvLdT6ZeYpzPEApk0E5tzivMW/VgpSdH
jWn0f84bcN5wGyDWbs2mAag8EtKpP6BrXruOIIt6keO1aO6g58QBdKhTCytKmg9l
Eg6CTY5j/e/rmxrbU6YTYK/CfdfHbBcl1HP7R2RQgYCUTOG/rksc35LtLgXfAgED
o1EwTzALBgNVHQ8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUJ/PI
FR5umgIJFq0roIlgX9p7L6owEAYJKwYBBAGCNxUBBAMCAQAwDQYJKoZIhvcNAQEF
BQADggEBAJ2dhISjQal8dwy3U8pORFBi71R803UXHOjgxkhLtv5MOhmBVrBW7hmW
Yqpao2TB9k5UM8Z3/sUcuuVdJcr18JOagxEu5sv4dEX+5wW4q+ffy0vhN4TauYuX
cB7w4ovXsNgOnbFp1iqRe6lJT37mjpXYgyc81WhJDtSd9i7rp77rMKSsH0T8lasz
Bvt9YAretIpjsJyp8qS5UwGH0GikJ3+r/+n6yUA4iGe0OcaEb1fJU9u6ju7AQ7L4
CYNu/2bPPu8Xs1gYJQk0XuPL1hS27PKSb3TkL4Eq1ZKR4OCXPDJoBYVL0fdX4lId
kxpUnwVwwEpxYB5DC2Ae/qPOgRnhCzU=
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIEPDCCAySgAwIBAgIKYQlufQAAAAAADDANBgkqhkiG9w0BAQUFADA1MRYwFAYD
VQQKEw1DaXNjbyBTeXN0ZW1zMRswGQYDVQQDExJDaXNjbyBSb290IENBIDIwNDgw
HhcNMTEwNjMwMTc1NjU3WhcNMjkwNTE0MjAyNTQyWjAnMQ4wDAYDVQQKEwVDaXNj
bzEVMBMGA1UEAxMMQUNUMiBTVURJIENBMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8A
MIIBCgKCAQEA0m5l3THIxA9tN/hS5qR/6UZRpdd+9aE2JbFkNjht6gfHKd477AkS
5XAtUs5oxDYVt/zEbslZq3+LR6qrqKKQVu6JYvH05UYLBqCj38s76NLk53905Wzp
9pRcmRCPuX+a6tHF/qRuOiJ44mdeDYZo3qPCpxzprWJDPclM4iYKHumMQmqmgmg+
xghHIooWS80BOcdiynEbeP5rZ7qRuewKMpl1TiI3WdBNjZjnpfjg66F+P4SaDkGb
BXdGj13oVeF+EyFWLrFjj97fL2+8oauV43Qrvnf3d/GfqXj7ew+z/sXlXtEOjSXJ
URsyMEj53Rdd9tJwHky8neapszS+r+kdVQIDAQABo4IBWjCCAVYwCwYDVR0PBAQD
AgHGMB0GA1UdDgQWBBRI2PHxwnDVW7t8cwmTr7i4MAP4fzAfBgNVHSMEGDAWgBQn
88gVHm6aAgkWrSugiWBf2nsvqjBDBgNVHR8EPDA6MDigNqA0hjJodHRwOi8vd3d3
LmNpc2NvLmNvbS9zZWN1cml0eS9wa2kvY3JsL2NyY2EyMDQ4LmNybDBQBggrBgEF
BQcBAQREMEIwQAYIKwYBBQUHMAKGNGh0dHA6Ly93d3cuY2lzY28uY29tL3NlY3Vy
aXR5L3BraS9jZXJ0cy9jcmNhMjA0OC5jZXIwXAYDVR0gBFUwUzBRBgorBgEEAQkV
AQwAMEMwQQYIKwYBBQUHAgEWNWh0dHA6Ly93d3cuY2lzY28uY29tL3NlY3VyaXR5
L3BraS9wb2xpY2llcy9pbmRleC5odG1sMBIGA1UdEwEB/wQIMAYBAf8CAQAwDQYJ
KoZIhvcNAQEFBQADggEBAGh1qclr9tx4hzWgDERm371yeuEmqcIfi9b9+GbMSJbi
ZHc/CcCl0lJu0a9zTXA9w47H9/t6leduGxb4WeLxcwCiUgvFtCa51Iklt8nNbcKY
/4dw1ex+7amATUQO4QggIE67wVIPu6bgAE3Ja/nRS3xKYSnj8H5TehimBSv6TECi
i5jUhOWryAK4dVo8hCjkjEkzu3ufBTJapnv89g9OE+H3VKM4L+/KdkUO+52djFKn
hyl47d7cZR4DY4LIuFM2P1As8YyjzoNpK/urSRI14WdIlplR1nH7KNDl5618yfVP
0IFJZBGrooCRBjOSwFv8cpWCbmWdPaCQT2nwIjTfY8c=
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIDezCCAmOgAwIBAgIEAc+JiTANBgkqhkiG9w0BAQsFADAnMQ4wDAYDVQQKEwVD
aXNjbzEVMBMGA1UEAxMMQUNUMiBTVURJIENBMB4XDTE3MDgxOTEwNDMzOVoXDTI3
MDgxOTEwNDMzOVowZzEmMCQGA1UEBRMdUElEOkM5MzAwLTI0VVggU046RkNXMjEz
NEwwMEMxDjAMBgNVBAoTBUNpc2NvMRgwFgYDVQQLEw9BQ1QtMiBMaXRlIFNVREEkx
EzARBgNVBAMTCkM5MzAwLTI0VVgwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK
AoIBAQDav5txv4THsqXwWC7AxzHm5MZ28Feqk8FA3tXAv0tV8RXtY4Z9I9XgRzwz
Yw8chknh8LuDMCmGmk8DP+ct++vAF4nkVeIeBeOHnx2RuC9rcR8tuKjCimamDk0M
JHk12w/9+TbdKdNBEy6Sueh1RPVbuSk1oQLQcOYW7CsYC5tI1GkJkfk1nGEK3ni3
ztIpsi7QHyp6k59yccnbzXSdwoBrtbpIIYEk/iHWFRQdlMUunnfIshI7yPneo7V0
NnPc08wk+CA+8XeXk/fnDeGAswKRK1tW9jDP/sY1YubBJNJ4ToqQpG6W/hbNvu3Y
NyS24osSvnn5Bp7on3Rf7eHq9hNjAgMBAAGjbzBtMA4GA1UdDwEB/wQEAwIF4DAM
BgNVHRMBAf8EAjAAME0GA1UdEQRGMESgQgYJKwYBBAEJFQIDoDUTM0NoaXBJRD1V
WUpPVkVZNEZRT0xSbkpwSUUxaGNpQXhNQ0F4Tnpvd05Eb3hNeUFiY1FjPTANBgkq
hkiG9w0BAQsFAAOCAQEASXX+iZLMvHQIR1/s1Pobm0kP/bYeHsgDTRQPRHbCM1HH
```

```
ROfjjDaJMHCspB17XtcLkNNFOwyUEkjoepyHjpxxhekGIqgD6Xt4rW6v/058Haw6
QbAhJFGZriVxFoBvW20VQ4ezyaGoqA+0I2GZqD/ZggUy6zsVwKmMe6inoEgXcYap
5GqF4weEoty9u+OKqr3ppWU475lxnNm/h+WHbNtunL6r7wZfe5dFQIxR5QP5gwRa
svpSsCoKm6PiwIUhw25CvtZ9NTg0tu5t5D7aVcxLeR8XbA1pjfgxw/RtSsjNse3+
ZkOgJUESqlxwzxcGUlY+vDINYRQ/sP6y7cT+niT00A==
-----END CERTIFICATE-----

Signature version: 1
Signature:
[redacted]
```

オプションの RSA 2048 署名は、3 つの証明書、署名のバージョンおよびユーザーにより提供されるナンスに対するものです。

```
RSA PKCS#1v1.5 Sign {<Nonce (UINT64)> || <Signature Version (UINT32)> || <Cisco Root CA
 2048 cert (DER)> ||
<Cisco subordinate CA (DER)> || <SUDI certificate (DER)> }
```

シスコの管理ソリューションには、上記の出力を解釈する機能が装備されています。しかし、OpenSSL コマンドを使用した簡単なスクリプトを使用してプラットフォームの ID を表示して署名を確認することもでき、それによってシスコの一意のデバイス ID を確保できます。

```
[linux-host:~]openssl x509 -in sudicert.pem -subject -noout
subject= /serialNumber=PID:C9300-24P SN:FDO1946BG05/O=Cisco/OU=ACT-2 Lite SUDI/CN=C9300-24P
```

ソフトウェア整合性の確認

次に、ブート段階のチェックサム レコードを表示する例を示します。ハッシュ測定値は、連続してブートされたソフトウェアの 3 つの段階それぞれについて表示されます。これらのハッシュをシスコが提供する基準値と比較できます。出力に署名するオプションを使用すると、出力は正規であり改ざんされていないことを保証する機能が検証ツールに付与されます。リプレイ攻撃から保護するために、ナンスを提供できます。

（注） ブート整合性ハッシュは MD5 ハッシュではありません。バンドルファイルに対して **verify /md5 cat9k_iosxe.16.10.01.SPA.bin** コマンドを実行すると、ハッシュは一致しません。

次に、インストールモードでの **show platform integrity sign nonce 123** コマンドの出力例を示します。この出力には、インストールされている各パッケージファイルの測定値が含まれます。

```
Device# show platform integrity sign nonce 123
Platform: C9300-24P
Boot 0 Version: MA0081R06.1307262016
Boot 0 Hash: A99EF9F31CE3F3F8533055407F1C88C62176E667E4E1DA0649EAA7A1282F205E0A
Boot Loader Version: System Bootstrap, Version 16.8.0.3, RELEASE SOFTWARE (P)
Boot Loader Hash:
F8282651468055C3993AB05F53512341BF20F3CC7D4083C980450EA6CD84608EE636B5B15D13414203CED35603F01974B867606A06F9DC45B25DD1039E686C40A
OS Version: 16.10.01
OS Hashes:
cat9k-cc_srdriver.16.10.01.SPA.pkg :
D0D155C1DEFDB03EB0C64057AD6A9673E2114FA7CCCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673E2114FA7CCCAAA7ED0AE935CB0BD84E0
cat9k-espbase.16.10.01.SPA.pkg :
3EB0C64057AD6A9673E2114FA7CCCAAA7ED0AE935CB0BD84E0D0D155C1DEFDB03EB0C64057AD6A9673E2114FA7CCCAAA7ED0AE935CB0BD84E0D0D155C1DEF43
cat9k-guestshell.16.10.01.SPA.pkg :
```

```
B0C64057AD6A9673E2114FA7CCCAAA7ED0AE935CB0BD84E0D0D155C1DEFDB03EB0C64057AD6A9673E2114FA7CCCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03E
cat9k-rpbase.16.10.01.SPA.pkg :
4057AD6A9673E2114FA7CCCAAA7ED0AE935CB0BD84E0D0D155C1DEFDB03EB0C64057AD6A9673E2114FA7CCCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C6
cat9k-rpboot.16.10.01.SPA.pkg :
AD6A9673E2114FA7CCCAAA7ED0AE935CB0BD84E0D0D155C1DEFDB03EB0C64057AD6A9673E2114FA7CCCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057
cat9k-sipbase.16.10.01.SPA.pkg :
9673E2114FA7CCCAAA7ED0AE935CB0BD84E0D0D155C1DEFDB03EB0C64057AD6A9673E2114FA7CCCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A
cat9k-sipspa.16.10.01.SPA.pkg :
E2114FA7CCCAAA7ED0AE935CB0BD84E0D0D155C1DEFDB03EB0C64057AD6A9673E2114FA7CCCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673
cat9k-srdriver.16.10.01.SPA.pkg :
4FA7CCCAAA7ED0AE935CB0BD84E0D0D155C1DEFDB03EB0C64057AD6A9673E2114FA7CCCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673E211
cat9k-webui.16.10.01.SPA.pkg :
CCCAAA7ED0AE935CB0BD84E0D0D155C1DEFDB03EB0C64057AD6A9673E2114FA7CCCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673E2114FA7
cat9k-wlc.16.10.01.SPA.pkg :
AA7ED0AE935CB0BD84E0D0D155C1DEFDB03EB0C64057AD6A9673E2114FA7CCCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673E2114FA7CCCA
PCR0: 9745B571B66D79F0936F4D292B5672B50F50FD1E56E74248D48A33582E992574
PCR8: 1CC295C233DA41BD3530A6F09C21991E8406BFFC88249D7778CA4BB0B9E71EB7
Signature version: 1
Signature:
[redacted]
```

次に、バンドルモードでの **show platform integrity sign nonce 123** コマンドの出力例を示します。この出力には、バンドルファイルとインストールされている各パッケージの測定値が含まれます。

```
Device# show platform integrity sign nonce 123
Platform: C9300-24P
Boot 0 Version: MA0081R06.1307262016
Boot 0 Hash: A99EF9F31CE3F3F8533055407F1C88C62176E667E4E1DA0649EAA7A1282F205E0A
Boot Loader Version: System Bootstrap, Version 16.8.0.3, RELEASE SOFTWARE (P)
Boot Loader Hash:
F828265146580E5C3993A895F53512341BF20F3CC7D4083C980450EA6CD84608EE636BEB15D13414203CED35603F61974B8676C6AC6F9DC45B25CD1039E686C49A
OS Version: 16.10.01
OS Hashes:
cat9k_iosxe.16.10.01.SPA.bin :
F4CAD08BE1EF841C3A2E3ED8540829F08F3CBA9336F38E45669D4D8B15AD15E365B922AC8B4DC0D5B63E2806D6A1BDAB7839DD9DC8CD7E366A49ED648C113440
cat9k-cc_srdriver.16.10.01.SPA.pkg :
D6D155C1DEFDB03EB0C64057AD6A9673E2114FA7CCCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673E2114FA7CCCAAA7ED0AE935CB0BD84E0
cat9k-espbase.16.10.01.SPA.pkg :
3EB0C64057AD6A9673E2114FA7CCCAAA7ED0AE935CB0BD84E0D0D155C1DEFDB03EB0C64057AD6A9673E2114FA7CCCAAA7ED0AE935CB0BD84E0D0D155C1DEF43
cat9k-guestshell.16.10.01.SPA.pkg :
B0C64057AD6A9673E2114FA7CCCAAA7ED0AE935CB0BD84E0D0D155C1DEFDB03EB0C64057AD6A9673E2114FA7CCCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03E
cat9k-rpbase.16.10.01.SPA.pkg :
4057AD6A9673E2114FA7CCCAAA7ED0AE935CB0BD84E0D0D155C1DEFDB03EB0C64057AD6A9673E2114FA7CCCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C6
cat9k-rpboot.16.10.01.SPA.pkg :
AD6A9673E2114FA7CCCAAA7ED0AE935CB0BD84E0D0D155C1DEFDB03EB0C64057AD6A9673E2114FA7CCCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057
cat9k-sipbase.16.10.01.SPA.pkg :
9673E2114FA7CCCAAA7ED0AE935CB0BD84E0D0D155C1DEFDB03EB0C64057AD6A9673E2114FA7CCCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A
cat9k-sipspa.16.10.01.SPA.pkg :
E2114FA7CCCAAA7ED0AE935CB0BD84E0D0D155C1DEFDB03EB0C64057AD6A9673E2114FA7CCCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673
cat9k-srdriver.16.10.01.SPA.pkg :
4FA7CCCAAA7ED0AE935CB0BD84E0D0D155C1DEFDB03EB0C64057AD6A9673E2114FA7CCCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673E211
cat9k-webui.16.10.01.SPA.pkg :
CCCAAA7ED0AE935CB0BD84E0D0D155C1DEFDB03EB0C64057AD6A9673E2114FA7CCCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673E2114FA7
cat9k-wlc.16.10.01.SPA.pkg :
AA7ED0AE935CB0BD84E0D0D155C1DEFDB03EB0C64057AD6A9673E2114FA7CCCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673E2114FA7CCCA
PCR0: 9745B571B66D79F0936F4D292B5672B50F50FD1E56E74248D48A33582E992574
PCR8: 1CC295C233DA41BD3530A6F09C21991E8406BFFC88249D7778CA4BB0B9E71EB7
Signature version: 1
Signature:
[redacted]
```

イメージ署名の検証

次に、SHA-512ハッシュを使用した、ブートアップ中のイメージに対するセキュアコード署名チェックの例を示します。

```
switch:boot flash:packages.conf
boot: attempting to boot from [flash:packages.conf]
boot: reading file packages.conf
#
Performing Integrity Check ...
boot: parsed image from conf file: cat9k-rpboot.17.02.01.SSA.pkg
##########################################################################################################################################################################################################################################################################################



Loading image in Verbose mode: 1



Image Base is: 0x100099000
Image Size is: 0x2C83487
Package header rev 3 structure detected
Package type:30001, flags:0x0
IsoSize = 0
Parsing package TLV info:
000: 000000090000001D4B45595F544C565F -         KEY_TLV_
010: 5041434B4147455F434F4D5041544942 - PACKAGE_COMPATIB
020: 494C4954590000000000000090000000B - ILITY
030: 4652555F52505F545950450000000009 - FRU_RP_TYPE
040: 000000184B45595F544C565F5041434B -     KEY_TLV_PACK
050: 4147455F424F4F544152434800000009 - AGE_BOOTARCH
060: 0000000E415243485F693638365F5459 -     ARCH_i686_TY
070: 5045000000000000900000144B45595F - PE          KEY_
080: 544C565F424F4152445F434F4D504154 - TLV_BOARD_COMPAT
090: 0000000900000010424F4152445F6361 -         BOARD_ca
0A0: 74396B5F545950450000000900000018 - t9k_TYPE
0B0: 4B45595F544C565F43525950544F5F4B - KEY_TLV_CRYPTO_K
0C0: 4559535452494E470000000900000004 - EYSTRING

TLV: T=9, L=29, V=KEY_TLV_PACKAGE_COMPATIBILITY
TLV: T=9, L=11, V=FRU_RP_TYPE
TLV: T=9, L=24, V=KEY_TLV_PACKAGE_BOOTARCH
TLV: T=9, L=14, V=ARCH_i686_TYPE
TLV: T=9, L=20, V=KEY_TLV_BOARD_COMPAT
TLV: T=9, L=16, V=BOARD_cat9k_TYPE
TLV: T=9, L=24, V=KEY_TLV_CRYPTO_KEYSTRING
TLV: T=9, L=4, V=none
TLV: T=9, L=11, V=CW_BEGIN=$$
TLV: T=9, L=17, V=CW_FAMILY=$cat9k$
TLV: T=9, L=74, V=CW_IMAGE=$cat9k-rpboot.17.02.01.SSA.pkg$
TLV: T=9, L=20, V=CW_VERSION=$17.2.01$
IOS version is 17.2.1
TLV: T=9, L=53, V=CW_FULL_VERSION=$17.2.01.0.869.1580816579..Amsterdam$
TLV: T=9, L=52, V=CW_DESCRIPTION=$Cisco IOS Software, IOS-XE Software$
TLV: T=9, L=9, V=CW_END=$$
Found DIGISIGN TLV type 12 length = 392
RSA Self Test Passed

Expected hash:
DDAF35A193617ABACC417349AE204131
12E6FA4E89A97EA20A9EEEE64B55D39A
```

```
2192992A274FC1A836BA3C23A3FEEBBD
454D4423643CE80E2A9AC94FA54CA49F

Obtained hash:
DDAF35A193617ABACC417349AE204131
12E6FA4E89A97EA20A9EEEE64B55D39A
2192992A274FC1A836BA3C23A3FEEBBD
454D4423643CE80E2A9AC94FA54CA49F
Sha512 Self Test Passed
Found package arch type ARCH_i686_TYPE
Found package FRU type FRU_RP_TYPE
Performing Integrity Check ...

RSA Signed DEVELOPMENT Image Signature Verification Successful.
```

ブート整合性の可視性に関する追加情報

関連資料

関連項目	マニュアル タイトル
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 9300* シリーズ スイッチ）

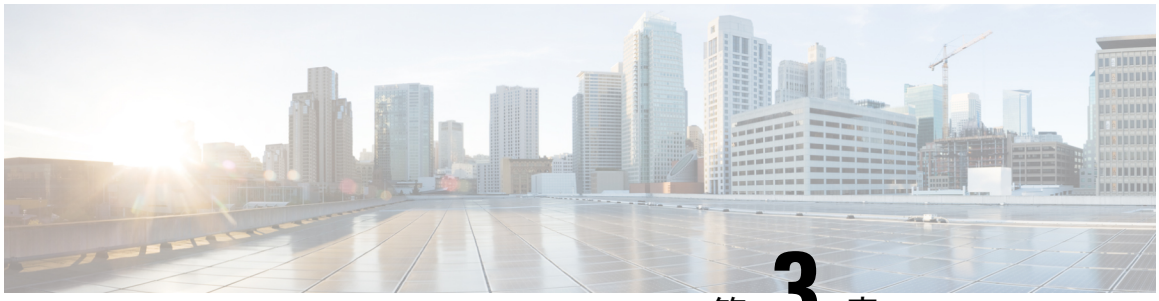
ブート整合性の可視性の機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

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

リリース	機能	機能情報
Cisco IOS XE Fuji 16.8.1a	ブート整合性の可視性	ブート整合性の可視性によって、シスコのプラットフォームIDとソフトウェアの整合性情報が可視化され、実用可能になります。プラットフォームIDは、プラットフォームの製造元でインストールされたIDを提供します。

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



第 3 章

デバイスのセットアップ設定の実行

デバイスセットアップの設定の制約事項

- サブパッケージソフトウェアのインストールはサポートされていません。

デバイスセットアップ設定の実行に関する情報

ここでは、IP アドレス割り当てと Dynamic Host Configuration Protocol（DHCP）の自動設定を含む、デバイスセットアップの設定方法について説明します。

デバイスブートプロセス

デバイスを起動するには、『*Cisco Catalyst 9300* シリーズ スイッチ ハードウェア設置ガイド』に記載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。このプロセスでは、物理メモリのマッピング場所、物理メモリの量と速度などを制御する CPU レジスタを初期化します。
- 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。
- 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。POST の一環として、次のテストも実行されます。
  - チップのアクセス可能性、ファームウェアのダウンロード、給電機器の正常性ステータスを確認する Power over Ethernet（PoE）コントローラの機能テスト。
  - デバイスセンサーからの温度の読み取りを確認する温度テスト。
  - 挿入されたすべてのファンモジュールがボード上で正常に動作しているかどうかを確認するファンモジュールテスト。
  - 連邦情報処理標準（FIPS）MACsec テスト。

  サポートされるオンライン診断の完全なリストについては、「オンライン診断の設定」の章を参照してください。

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
```

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

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

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

バンドル モードでのブート

バンドル（.bin）ファイルを使用して、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

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

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

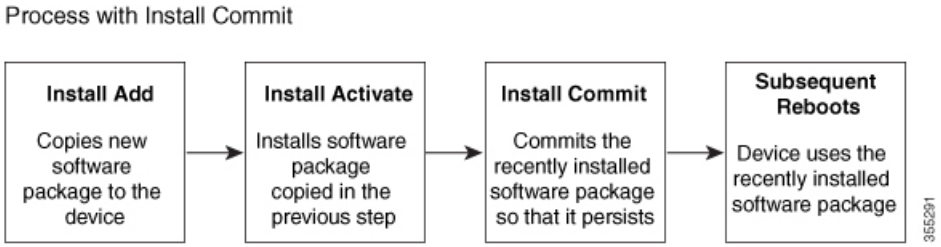
install activate コマンドを動作させるには、パッケージをデバイスのブートフラッシュで使用可能にする必要があります。このコマンドを設定すると、.bin ファイルから以前に追加したパッケージがアクティブ化され、システムがリロードします。

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

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

次のフロー チャートで、ソフトウェアのインストールの動作を説明します。

図 *3:* ソフトウェア パッケージのコミット



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

ソフトウェアインストールの終了

ソフトウェアイメージのアクティブ化は次の方法で終了できます。

- **install activate auto-abort-timer** コマンドを使用します。新しいイメージをアクティブ化した後にデバイスをリロードすると、auto-abort-timer がトリガーされます。**install commit** コマンドを発行する前にタイマーが期限切れになった場合、インストールプロセスが終了します。デバイスは再度リロードし、前のバージョンのソフトウェアイメージで起動します。

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

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

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、DHCP サーバーを使用する方法、または手動で実行する方法があります。

特定の IP 情報の設定が必要な場合、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。

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

（注） DHCP を使用している場合は、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ーの場合は、デバイスを手動で設定してください。それ以外のユーザーは、デバイスブートプロセス （55 ページ） のセクション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 *4:* デフォルトのスイッチ情報

機能	デフォルト設定
IP アドレスおよびサブネット マスク	IP アドレスまたはサブネット マスクは定義されていません。
デフォルト ゲートウェイ	デフォルト ゲートウェイは定義されていません。
イネーブル シークレット パスワード	パスワードは定義されていません。
ホスト名	出荷時に割り当てられるデフォルトのホスト名は device 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 サーバー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/サーバー モデルに基づいています。指定された DHCP サーバーが、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 サーバー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 サーバーで、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サーバーおよびドメイン ネーム システム（DNS）サーバーの設定が必要になることがあります。

デバイスの DHCP サーバーは、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 サーバーが異なる LAN 上で動作している場合、デバイスとDHCPサーバー間に、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。

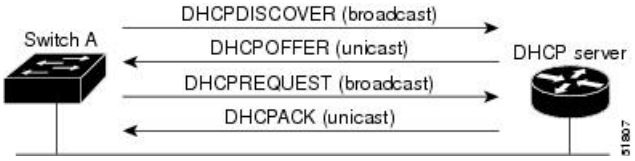
DHCP ベースの自動設定は、デバイスの BOOTP クライアント機能に代わるものです。

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

次は、DHCP クライアントと DHCP サーバー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/サーバー間のメッセージ交換



クライアントであるデバイスAは、DHCPサーバーの場所を特定するために、DHCPDISCOVER メッセージをブロードキャストします。DHCP サーバーは、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サーバー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 サーバーは、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 サーバーは、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ーはバウンドされ、クライアントはサーバー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 サーバー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 サーバーに、DHCPDECLINE ブロードキャスト メッセージを戻します。

DHCP サーバーはクライアントに、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 サーバー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 サーバーまたは BOOTP サーバー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 サーバーから提示された 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、サーバーは通常、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 サーバー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 サーバーから取得できます。クライアント（デバイス）は DCHPDISCOVER メッセージ内に、DHCP サーバーからのホスト名および他のコンフィギュレーション パラメータの要求に使用される Option 12 フィールドを加えます。すべてのクライアントのコンフィギュレーション ファイルは、DHCP から取得したホスト名を除き、まったく同じです。

DHCP ベースの自動設定およびイメージ アップデート

DHCP イメージアップグレード機能を使用すると、ネットワーク内の 1 つ以上の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 サーバーを設定できます。ネットワーク内のすべてのスイッチでのイメージおよびコンフィギュレーションの同時アップグレードによって、ネットワークに加えられたそれぞれの新しいデバイスが、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 **copyrunning-configuration startup-configuration** 特権 EXEC コマンドを入力しない限り、NVRAMに保存されません。ダウンロードされたコンフィギュレーションがスタートアップ コンフィギュレーションに保存された場合、後続のシステム再起動中にこの機能はトリガーされません。

DHCP 自動設定

DHCP 自動設定は、コンフィギュレーション ファイルを DHCP サーバーからネットワーク内の 1 つ以上の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サーバーを、正しいオプション67（コンフィギュレーション ファイル名）、オプション 66（DHCP サーバーホスト名）、オプション 150（TFTP サーバーアドレス）、およびオプション 125（Cisco IOS イメージファイルの説明）の設定で設定する必要があります。

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 サーバー設定時の注意事項

デバイスを DHCP サーバーとして設定する場合、次の注意事項に従ってください。

- DHCPサーバーには、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 サーバーに次のリースオプションを設定する必要があります。
  - クライアントの IP アドレス（必須）
  - クライアントのサブネット マスク（必須）
  - DNS サーバーの 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 サーバーからコンフィギュレーション ファイルを受信させる場合は、DHCP サーバーに次のリースオプションを設定する必要があります。
  - TFTP サーバー名（必須）
  - ブートファイル名（クライアントが必要とするコンフィギュレーションファイル名）（推奨）
  - ホスト名（任意）
- DHCP サーバーの設定によっては、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 サーバーは、設定されたパラメータのみを使用してクライアントの要求に応答します。IP アドレスおよびサブネットマスクが応答に含まれていないと、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 サーバー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 サーバーとして動作することができます。デフォルトでは、Cisco IOS DHCP サーバーおよび DHCP リレーエージェント機能は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 サーバーの目的

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

コンフィギュレーションファイル名、および TFTP サーバーを指定しなかった場合、またはコンフィギュレーションファイルをダウンロードできなかった場合は、デバイスはファイル名と TFTP サーバーアドレスをさまざまに組み合わせてコンフィギュレーション ファイルをダウンロードしようとします。ファイルには、特定のコンフィギュレーションファイル名（存在する場合）と次のファイルが指定されています。network-config、cisconet.cfg、*hostname*.config、ま

たは *hostname*.cfg です。この場合、*hostname* はデバイスの現在のホスト名です。使用される TFTP サーバーアドレスには、（存在する場合）指定された TFTP サーバーのアドレス、およびブロードキャスト アドレス（255.255.255.255）が含まれています。

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 サーバーのベースディレクトリに1つまたは複数のコンフィギュレーション ファイルが含まれていなければなりません。含めることのできるファイルは、次のとおりです。

- DHCP応答で指定されているコンフィギュレーション ファイル（実際のデバイス 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 サーバー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 サーバー リース データベースに TFTP サーバー名を指定する場合は、DNS サーバーのデータベースに TFTP サーバー名と IP アドレスのマッピングを設定することも必要です。

使用するTFTPサーバーが、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サーバーの応答に含まれていない場合に発生）は、リレーを設定して TFTP サーバーに TFTP パケットを転送する必要があります。適切な解決方法は、必要なすべての情報を使用して DHCP サーバーを設定することです。

DNS サーバーの目的

DHCP サーバーは、DNS サーバーを使用して TFTP サーバー名を IP アドレスに変換します。DNSサーバー上で、TFTPサーバー名から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 サーバーには、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 サーバーのリース データベースに、DNS サーバーの IP アドレスを設定できます。リース データベースには、DNS サーバーの IP アドレスを 2 つまで入力できます。

DNS サーバーは、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サーバーが別のLAN上に存在する場合、デバイスはルータを介してDNSサーバー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サーバーから、IPアドレス、サブネットマスク、TFTPサーバーアドレス、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サーバー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ー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サーバー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サーバーから、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サーバー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 ファイルをサーバー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サーバーから、IPアドレス、サブネットマスク、およびTFTPサーバーアドレスを受信します。デバイスは、TFTPサーバー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*Switch*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*.cf）をTFTPサーバーから読み込みます。cisconet.cfgファイルが読み込まれている場合は、ホストのファイル名は8文字に切り捨てられます。

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

環境変数の制御方法

通常動作デバイスでは、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 がグリーンの点滅から点灯したままになったら、[Mode] ボタンを放してもかまいません。ブートローダの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、ブートローダー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 *5:* 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。たとえば、**boot flash:packages.conf** です。 • 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。たとえば、**boot flash:image.bin** のようになります。

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 *6 : 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_GATEWAY	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、週末などデバイスをあまり使用しないときに）行うように、スケジュールを設定できます。または（ネットワーク内のすべてのデバイスでソフトウェアをアップグレードする場合など）ネットワーク全体でリロードを同時に行うことができます。

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

リロード オプションには以下のものがあります。

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

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

手動で起動するように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ーが制御を失う事態を防止するための制約です。

コンフィギュレーションファイルを変更すると、リロードの前にコンフィギュレーションを保存するように指示するプロンプトが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 サーバーおよび TFTP サーバー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 の設定の 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 サーバー 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 サーバーの 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 サーバー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 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 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。このテキストファイル内に、ダウンロードするイメージの名前を含めます（たとえば、cat9k_iosxe.16.xx.xx.SPA.bin）。

手順

	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 サーバー 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 サーバーの 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 0000.0009.0a05.0861.7574.6f69.6e73.7461.6c6c.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 サーバー上の 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 サーバー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	**copy running-config startup-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 サーバー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サーバーから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	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル設定モードを開始します。
ステップ 3	**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 4	**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 5	**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 6	**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。

	コマンドまたはアクション	目的
		（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。 （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 7	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 8	**show interfaces vlan** *vlan-id* 例： `Device# show interfaces vlan 99`	指定した VLAN のインターフェイスステータスを表示します。
ステップ 9	**show ip redirects** 例： `Device# show ip redirects`	Internet Control Message Protocol（ICMP）リダイレクトメッセージを表示します。

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

次のセクションでは、デバイスのスタートアップ コンフィギュレーションを変更する方法について説明します。

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	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル設定モードを開始します。
ステップ 3	**boot flash**:*/file-url* 例： `Device(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 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show boot** 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。 • **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 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。

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。
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** グローバルコマンドは、MANUAL_BOOT 環境変数の設定を変更します。 次回、システムを再起動した際には、スイッチはブートローダ モードになり、ブートローダ モードであることが *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:`

	コマンドまたはアクション	目的
		• *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	**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。

インストール モードでのデバイスのブート

ソフトウェア パッケージのインストール

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

手順

	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 モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**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。

	コマンドまたはアクション	目的
		• このコマンドの実行後にデバイスはリロードします。
ステップ 3	**exit** 例： `Device# exit`	特権 EXEC モードを終了し、ユーザー EXEC モードに戻ります。

更新プログラム パッケージの管理

手順

	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 モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**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

	コマンドまたはアクション	目的
		場合、インストールプロセスは自動的に終了します。デバイスがリロードし、以前のバージョンのソフトウェア イメージで起動します。
ステップ 4	**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終了し、現在のインストール手順の前に実行していたバージョンにロールバックします。 • 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： `Device# install commit`	リロードが繰り返されても持続する変更を行います。 • **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。

バンドルモードでのデバイスの起動

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 サーバーから 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 サーバー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:` `switch:BOOT=tftp://10.0.0.2/cat9k_iosxe.16.05.01a.SPA.bin` `switch:` `switch:`	ブート パラメータを設定します。
ステップ 2	**boot** 例： `switch:boot`	デバイスを起動します。
ステップ 3	**show version**	（任意）インストールされているイメージのバージョンを表示します。

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	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル設定モードを開始します。
ステップ 3	**copy running-config startup-config** 例： `Device# 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。

	コマンドまたはアクション	目的
ステップ 4	**reload in** [*hh*:]*mm* [*text*] 例： `Device# 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 5	**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） **at** キーワードを使用するのは、デバイスのシステムクロックが（Network Time Protocol（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だけです。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 6	**reload cancel** 例： `Device(config)# reload cancel`	以前にスケジューリングされたリロードをキャンセルします。
ステップ 7	**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の設定例

次のセクションにデバイスセットアップの設定例を示します。

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

次の例では、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!
```

次の例では、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

次に、**emergency-install** ブートコマンドの出力例を示します。

```
switch: emergency-install tftp://10.10.0.10/auto/tftpboot/X86/cat9k_iosxe.16.05.01a.SPA.bin
WARNING: The system partition (bootflash:) will be erased during the system recovery
install process.
Are you sure you want to proceed? [y]  y/n [n]: y
Starting system recovery
(tftp://210.10.0.10/auto/tftpboot/X86/cat9k_iosxe.16.05.01a.SPA.bin) ...
Attempting to boot from [sda9:cat9k-recovery.SSA.bin]
Located cat9k-recovery.SSA.bin
############################################################################################################################

Warning: ignoring ROMMON var "BOOT_PARAM"

PLATFORM_TYPE C9300 speed 9600

Booting Recovery Image 16.5.1a



Initiating Emergency Installation of bundle
tftp://10.10.0.10/auto/tftpboot/X86/cat9k_iosxe.16.05.01a.SPA.bin


Downloading bundle tftp://10.10.0.10/auto/tftpboot/X86/cat9k_iosxe.16.05.01a.SPA.bin...
curl_vrf=2
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  485M  100  485M    0     0  5143k      0  0:01:36  0:01:36 --:--:-- 5256k
100  485M  100  485M    0     0  5143k      0  0:01:36  0:01:36 --:--:-- 5143k

Validating bundle tftp://10.10.0.10/auto/tftpboot/X86/cat9k_iosxe.16.05.01a.SPA.bin...
Installing bundle tftp://10.10.0.10/auto/tftpboot/X86/cat9k_iosxe.16.05.01a.SPA.bin....
Verifying bundle tftp://10.10.0.10/auto/tftpboot/X86/cat9k_iosxe.16.05.01a.SPA.bin...
Package cat9k-cc_srdriver.16.05.01a.SPA.pkg
/temp//stage/cat9k-cc_srdriver.16.05.01a.SPA.pkg is Digitally Signed
Package cat9k-espbase.16.05.01a.SPA.pkg /temp//stage/cat9k-espbase.16.05.01a.SPA.pkg is
 Digitally Signed
Package cat9k-guestshell.16.05.01a.SPA.pkg /temp//stage/cat9k-guestshell.16.05.01a.SPA.pkg
 is Digitally Signed
Package cat9k-rpbase.16.05.01a.SPA.pkg /temp//stage/cat9k-rpbase.16.05.01a.SPA.pkg is
Digitally Signed
Package cat9k-sipbase.16.05.01a.SPA.pkg /temp//stage/cat9k-sipbase.16.05.01a.SPA.pkg is
 Digitally Signed
Package cat9k-sipspa.16.05.01a.SPA.pkg /temp//stage/cat9k-sipspa.16.05.01a.SPA.pkg is
Digitally Signed
Package cat9k-srdriver.16.05.01a.SPA.pkg /temp//stage/cat9k-srdriver.16.05.01a.SPA.pkg
is Digitally Signed
Package cat9k-webui.16.05.01a.SPA.pkg /temp//stage/cat9k-webui.16.05.01a.SPA.pkg is
Digitally Signed
Package cat9k-wlc.16.05.01a.SPA.pkg /temp//stage/cat9k-wlc.16.05.01a.SPA.pkg is Digitally
 Signed
Package /cat9k-rpboot.16.05.01a.SPA.pkg /temp//rpboot/cat9k-rpboot.16.05.01a.SPA.pkg is
 Digitally Signed
Preparing flash....
Flash filesystem unmounted successfully /dev/sdb3
Syncing device....
```

```
Emergency Install successful... Rebooting
Will reboot now

Initializing Hardware...

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

Current image running:
Primary Rommon Image

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

例：更新プログラム パッケージの管理

次に、ソフトウェア パッケージ ファイルを追加する例を示します。

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

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

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

[OK]Modified configuration has been saved

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

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

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

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

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

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

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

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

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

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

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

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

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

 Chassis 1 reloading, reason - Reload command
```

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

```
Device# show install summary

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

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

```
Device# install activate

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

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

%INSTALL-5-INSTALL_START_INFO: Started install activateFollowing packages shall be
activated:
/flash/cat9k-wlc.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
/flash/cat9k-webui.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
```

```
/flash/cat9k-srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
/flash/cat9k-sipspa.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
/flash/cat9k-sipbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
/flash/cat9k-rpboot.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
/flash/cat9k-rpbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
/flash/cat9k-guestshell.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
/flash/cat9k-espbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
/flash/cat9k-cc_srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg

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


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

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

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

                                     Chassis 1 reloading, reason - Reload command
```

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

```
Device# show install summary

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

```
           C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
IMG   I    16.6.2.0
IMG   U    16.6.1.0
Device#
```

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

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

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

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

Device#
```

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

```
Device# install rollback to committed

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

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

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

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

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

```
    Replacement:   cc 8 0   cc              cat9k-sipbase.16.06.02.SPA.pkg
    Replacement:   cc 8 0   cc_spa          cat9k-sipspa.16.06.02.SPA.pkg
    Replacement:   cc 9  0  cc_srdriver     cat9k-cc_srdriver.16.06.02.SPA.pkg
    Replacement:   cc 9 0   cc              cat9k-sipbase.16.06.02.SPA.pkg
    Replacement:   cc 9 0   cc_spa          cat9k-sipspa.16.06.02.SPA.pkg
    Replacement:   fp 0 0   fp              cat9k-espbase.16.06.02.SPA.pkg
    Replacement:   fp 1 0   fp              cat9k-espbase.16.06.02.SPA.pkg
    Replacement:   rp 0 0   guestshell      cat9k-guestshell.16.06.02.SPA.pkg
    Replacement:   rp 0 0   rp_base         cat9k-rpbase.16.06.02.SPA.pkg
    Replacement:   rp 0 0   rp_daemons      cat9k-rpbase.16.06.02.SPA.pkg
    Replacement:   rp 0 0   rp_iosd         cat9k-rpbase.16.06.02.SPA.pkg
    Replacement:   rp 0 0   rp_security     cat9k-rpbase.16.06.02.SPA.pkg
    Replacement:   rp 0 0   rp_webui        cat9k-webui.16.06.02.SPA.pkg
    Replacement:   rp 0 0   rp_wlc          cat9k-wlc.16.06.02.SPA.pkg
    Replacement:   rp 0 0   srdriver        cat9k-srdriver.16.06.02.SPA.pkg
    Replacement:   rp 1 0   guestshell      cat9k-guestshell.16.06.02.SPA.pkg
    Replacement:   rp 1 0   rp_base         cat9k-rpbase.16.06.02.SPA.pkg
    Replacement:   rp 1 0   rp_daemons      cat9k-rpbase.16.06.02.SPA.pkg
    Replacement:   rp 1 0   rp_iosd         cat9k-rpbase.16.06.02.SPA.pkg
    Replacement:   rp 1 0   rp_security     cat9k-rpbase.16.06.02.SPA.pkg

    Chassis 1 reloading, reason - Reload command


    Replacement:   rp 1 0   rp_webui        cat9k-webui.16.06.02.SPA.pkg
    Replacement:   rp 1 0   rp_wlc          cat9k-wlc.16.06.02.SPA.pkg
    Replacement:   rp 1 0   srdriver        cat9k-srdriver.16.06.02.SPA.pkg
    Finished rollback impact
  [1] Finished Rollback on switch 1
Checking status of Rollback on [1]
Rollback: Passed on [1]
Finished Rollback

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

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

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

```
Device# install remove inactive

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

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

```
/flash/cat9k-webui.16.06.02.SPA.pkg
/flash/cat9k-wlc.16.06.02.SPA.pkg
/flash/packages.conf

Do you want to remove the above files? [y/n]y
[switch 1]:
Deleting file flash:cat9k-cc_srdriver.16.06.02.SPA.pkg ... done.
Deleting file flash:cat9k-espbase.16.06.02.SPA.pkg ... done.
Deleting file flash:cat9k-guestshell.16.06.02.SPA.pkg ... done.
Deleting file flash:cat9k-rpbase.16.06.02.SPA.pkg ... done.
Deleting file flash:cat9k-rpboot.16.06.02.SPA.pkg ... done.
Deleting file flash:cat9k-sipbase.16.06.02.SPA.pkg ... done.
Deleting file flash:cat9k-sipspa.16.06.02.SPA.pkg ... done.
Deleting file flash:cat9k-srdriver.16.06.02.SPA.pkg ... done.
Deleting file flash:cat9k-webui.16.06.02.SPA.pkg ... done.
Deleting file flash:cat9k-wlc.16.06.02.SPA.pkg ... done.
Deleting file flash:packages.conf ... done.
SUCCESS: Files deleted.
--- Starting Post_Remove_Cleanup ---
Performing Post_Remove_Cleanup on all members
  [1] Post_Remove_Cleanup package(s) on switch 1
  [1] Finished Post_Remove_Cleanup on switch 1
Checking status of Post_Remove_Cleanup on [1]
Post_Remove_Cleanup: Passed on [1]
Finished Post_Remove_Cleanup

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

次に、**install abort** コマンドの出力例を示します。

```
Device# install abort

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

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

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

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

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

```
    Replacement:   cc 8 0   cc_spa          cat9k-sipspa.16.06.02.SPA.pkg
    Replacement:   cc 9  0  cc_srdriver     cat9k-cc_srdriver.16.06.02.SPA.pkg
    Replacement:   cc 9 0   cc              cat9k-sipbase.16.06.02.SPA.pkg
    Replacement:   cc 9 0   cc_spa          cat9k-sipspa.16.06.02.SPA.pkg
    Replacement:   fp 0 0   fp              cat9k-espbase.16.06.02.SPA.pkg
    Replacement:   fp 1 0   fp              cat9k-espbase.16.06.02.SPA.pkg
    Replacement:   rp 0 0   guestshell      cat9k-guestshell.16.06.02.SPA.pkg
    Replacement:   rp 0 0   rp_base         cat9k-rpbase.16.06.02.SPA.pkg
    Replacement:   rp 0 0   rp_daemons      cat9k-rpbase.16.06.02.SPA.pkg
    Replacement:   rp 0 0   rp_iosd         cat9k-rpbase.16.06.02.SPA.pkg
    Replacement:   rp 0 0   rp_security     cat9k-rpbase.16.06.02.SPA.pkg
    Replacement:   rp 0 0   rp_webui        cat9k-webui.16.06.02.SPA.pkg
    Replacement:   rp 0 0   rp_wlc          cat9k-wlc.16.06.02.SPA.pkg
    Replacement:   rp 0 0   srdriver        cat9k-srdriver.16.06.02.SPA.pkg
    Replacement:   rp 1 0   guestshell      cat9k-guestshell.16.06.02.SPA.pkg
    Replacement:   rp 1 0   rp_base         cat9k-rpbase.16.06.02.SPA.pkg
    Replacement:   rp 1 0   rp_daemons      cat9k-rpbase.16.06.02.SPA.pkg
    Replacement:   rp 1 0   rp_iosd         cat9k-rpbase.16.06.02.SPA.pkg
    Replacement:   rp 1 0   rp_security     cat9k-rpbase.16.06.02.SPA.pkg
    Replacement:   rp 1 0   rp_webui        cat9k-webui.16.06.02.SPA.pkg
    Replacement:   rp 1 0   rp_wlc          cat9k-wlc.16.06.02.SPA.pkg
    Replacement:   rp 1 0   srdriver        cat9k-srdriver.16.06.02.SPA.pkg
    Finished rollback impact
  [1] Finished Abort on switch 1
Checking status of Abort on [1]
Abort: Passed on [1]
Finished Abort

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

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

```
Device# install activate auto-abort-timer 30

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

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

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

This operation requires a reload of the system. Do you want to proceed? [y/n]y
--- Starting Activate ---
Performing Activate on all members
  [1] Activate package(s) on switch 1
    --- Starting list of software package changes ---
    Old files list:
```

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


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

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

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

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

ソフトウェア インストールの確認

手順

ステップ 1 **enable**

例：

```
Device> enable
```

特権 EXEC モードを有効にします。

- パスワードを入力します（要求された場合）。

ステップ 2 **show install log**

例：

```
Device# show install log
```

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

```
Device# show install log

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

ステップ 3 **show install summary**

例：

```
Device# show install summary
```

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

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

```
Device# show install summary

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

Device#
```

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

例：

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

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

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

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

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

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

  Package is not bootable.
Device#
```

ステップ 5 show install active

例：

```
Device# show install active
```

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

```
Device# show install active

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

ステップ 6 show install inactive

例：

```
Device# show install inactive
```

非アクティブなパッケージに関する情報を表示します。

```
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：デバイスを DHCP サーバーとして設定

```
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 サーバーから設定をダウンロードするためのデバイスの設定

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

```
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]
```

デバイスセットアップの実行に関する追加情報

関連資料

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

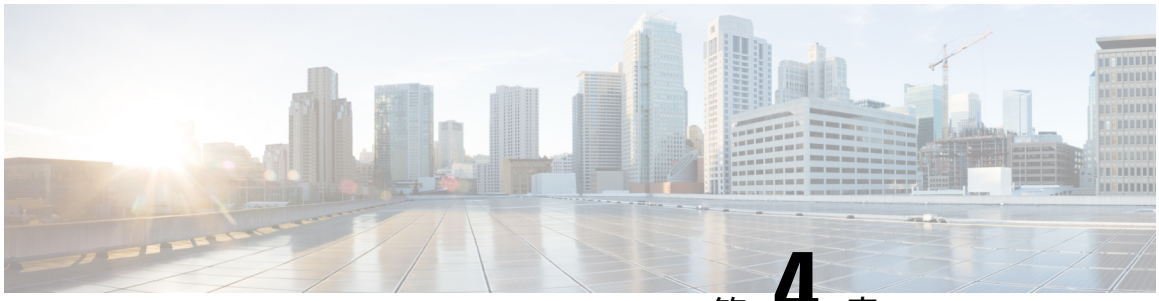
デバイスセットアップ設定の実行に関する機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

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

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

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



第 4 章

使用可能なライセンス

使用可能なライセンスに関する情報

ここでは、Cisco IOS-XE ソフトウェアを実行している Cisco Catalyst 9300 シリーズ スイッチで使用可能なライセンスについて説明します。特に指定のない限り、この情報はシリーズのすべてのモデルに適用されます。

基本ライセンスとアドオンライセンス

次の基本ライセンスとアドオンライセンスを使用できます。

基本ライセンス

基本ライセンスとは、永続的に有効な永久ライセンスです。こうしたライセンスには使用期限日はありません。

- Network Essentials
- Network Advantage：Network Essentials ライセンスで使用可能な機能と追加機能が含まれます。

アドオン ライセンス

アドオンライセンスでは、スイッチだけでなく Cisco Digital Network Architecture Center（Cisco DNA Center）でもシスコのイノベーションを提供しています。

アドオンライセンスは特定の日付まで有効です。アドオンライセンスは 3 年、5 年、または 7 年のサブスクリプション期間にわたって購入できます。

- DNA Essentials

- DNA Advantage：DNA Essentials ライセンスで使用可能な機能と追加機能が含まれます。

基本ライセンスとアドオンライセンスの使用に関するガイドライン

- 基本ライセンス（Network Essentials および Network-Advantage）の注文および履行は、無期限または永久ライセンスタイプのみとなります。
- アドオンライセンス（DNA Essentials および DNA Advantage）の注文および履行は、サブスクリプションまたは有効期間付きライセンスタイプのみとなります。
- ネットワーク ライセンス レベルを選択した場合はアドオンライセンスレベルが含まれています。DNA 機能を使用する場合は、期限が切れる前にライセンスを更新して、使用を継続してください。DNA 機能の使用を継続しない場合は、アドオンライセンスを非アクティブ化してからスイッチをリロードして基本ライセンス機能での運用を継続します。

  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 *7*: 表 *4* 許可されている組み合わせ

	DNA Essentials	DNA Advantage
Network Essentials	対応	非対応
Network Advantage	可[1]	対応

[1] この組み合わせは DNA ライセンスの更新時にのみ購入できます。DNA-Essentials の初回購入時には購入できません。

- 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、https://cfnng.cisco.com に進みます。cisco.com のアカウントは必要ありません。

高セキュリティライセンス

暗号化機能を提供する製品および機能は、米国輸出管理法、米国政府暗号化および輸出管理規則（EAR）の範囲内です。[2]高セキュリティ（HSECK9）は、輸出規制対象のライセンスであり、暗号化機能の使用を許可します。

このサブセクションでは、ライセンスをサポートする製品、ライセンスを必要とする暗号化機能、ライセンスを注文する際の考慮事項、ライセンスを使用するための前提条件、およびサポートされるプラットフォームでのライセンスの設定方法について説明します。

[2]

サポートされているプラットフォームとリリース

HSECK9 ライセンスは、Cisco IOS XE Bengaluru 17.6.2 以降の Cisco Catalyst 9300X シリーズ スイッチでのみ使用できます。

シリーズで使用可能な SLU の詳細については、『Cisco Catalyst 9300 Series Switches Hardware Installation Guide』を参照してください。

HSECK9 ライセンスが必要な場合

HSECK9 ライセンスは、米国の輸出規制法の制限対象である、特定の暗号化機能を使用する場合にのみ必要です。HSECK9 ライセンスがないと、制限対象の暗号化機能を有効にできません。

IPsec 機能には HSECK9 ライセンスが必要です。

HSECK9 ライセンスを使用するための前提条件

次の要件を満たしていることを確認します。

- デバイスが HSECK9 ライセンスをサポートしていること。サポートされているプラットフォームとリリース （111 ページ）を参照してください。
- Cisco Smart Software Manager（CSSM）の該当するスマートアカウントおよびバーチャルアカウントに必要な数の HSECK9 ライセンスがあること。暗号化機能を使用する UDI ごとに、HSECK9 ライセンスが 1 つ必要です。必要なライセンス数に関しては、スタック構成の考慮事項を確認してください。スタッキングに関する考慮事項 （112 ページ）を参照してください。
- サポートされている Smart Licensing Using Policy トポロジのいずれかを実装していること。これにより、使用する HSECK9 ライセンスごとにスマートライセンス承認コード（SLAC）をインストールできます。

  HSECK9 ライセンスは、米国の取引規制法（輸出規制）の制限対象であるため、使用前に承認が必要です。SLAC はこの承認を提供し、輸出規制対象のライセンスの有効化と継続的な使用を可能にします。SLAC は CSSM で生成され、CSSM から取得されます。デバイスを CSSM に接続して SLAC を取得する方法はいくつかあります。CSSM に接続する各方法がトポロジと呼ばれます。設定セクションは、各トポロジで SLAC を取得する方法を示します（HSECK9 ライセンス用の SLAC のインストール （115 ページ））。

（注） このドキュメント（サポートされているプラットフォームとリリース （111 ページ））の範囲内にあるサポート対象プラットフォームで SLAC を取得してインストールするには、このドキュメントの設定セクションを参照してください。他のシスコ製品と比較すると、設定プロセスに違いがあります。

- 暗号化機能の設定は、SLAC をインストールしてから行います。インストール前に暗号化機能を設定した場合、SLAC のインストール後に再設定する必要があります。

発注時の考慮事項

ここでは、HSECK9 ライセンスの発注に関する重要な考慮事項について説明します。

暗号化機能を使用する UDI ごとに、個別の HSECK9 ライセンスが必要です。デバイススタックがある場合は、スタッキングに関する考慮事項（112ページ）セクションで必要なライセンス数に関する情報を参照してください。

注文する新しいハードウェア（サポートされているプラットフォーム）で暗号化機能を使用する予定の場合は、スマートアカウントとバーチャルアカウントの情報を注文時に提供します。これにより、SLAC を工場でインストールできます。

ライセンスの注文については、『Cisco Catalyst 9300 Series Ordering Guide』を参照してください。https://www.cisco.com/c/en/us/products/collateral/switches/catalyst-9300-series-switches/nb-06-cat9300-swit-ser-cte-en.html

スタッキングに関する考慮事項

このセクションでは、アクティブ、スタンバイ、および1つ以上のメンバーを持つデバイススタックに適用される HSECK9 ライセンスの考慮事項と要件について説明します。

- 混合スタック構成はサポートされていません。

  スタック内のすべてのデバイスは、Cisco Catalyst 9300X シリーズ スイッチである必要があります。

- 最低限、HSECK9 ライセンスを取得し、スタック内のアクティブデバイスの SLAC をインストールする必要があります。スイッチオーバー時に暗号化機能を中断なく使用するため、スタンバイ用の HSECK9 ライセンスも取得することを推奨します。

  スイッチオーバーが発生し、スタンバイに HSECK9 ライセンスがない場合、暗号化機能は無効になります。システムは自動的にデバイススタックをリロードし、スタック全体で暗号化機能を無効にします。

  - デバイススタックのスタンバイに HSECK9 ライセンスがインストールされていない場合に表示される、毎日のシステムメッセージ。これは、スイッチオーバーが発生したときに暗号化機能が無効になることのみを警告するものです。現在アクティブなデバイスの HSECK9 対応機能の動作には影響しません。

    ```
    %IOSXE_SMART_AGENT-6-STANDBY_NOT_AUTHORIZED: Standby is in 'not authorized' state
     for license hseck9.
    ```

  - スイッチオーバーが発生したときに表示されるシステムメッセージ。この場合、スタンバイに HSECK9 ライセンスがありません。これらのメッセージは、デバイスがリロードされていることを警告します。リロード後にシステムが起動すると、暗号化機能はスタック全体で無効になります。

    ```
    %PLATFORM_IPSEC_HSEC-3-UNAUTHORIZED_HSEC: Switchover happened with IPSec
    configured but
    HSEC unauthorized, reloading.
    ```

```
%PMAN-5-EXITACTION: F0/0: pvp: Process manager is exiting: reload fp action
requested
%PMAN-5-EXITACTION: R0/0: pvp: Process manager is exiting: rp processes exit
with reload switch code
```

- 暗号化機能がすでに使用されている既存のスタックにデバイスを追加するには、次のいずれかの手順を実行します。
  - SLAC をインストールし、スタンドアロンデバイスで暗号化機能を設定し、最後に既存のスタックにデバイスを追加します。
  - デバイスをスタックに追加し、スタック全体の SLAC を再度要求します。

使用可能なライセンスの設定方法

ここでは、使用可能なライセンスの設定方法と、ライセンスを設定する前後に必要なタスクについて説明します。

基本ライセンスとアドオンライセンスの設定

基本ライセンスまたはアドオンライセンスを注文および購入したら、使用する前にデバイスでライセンスを設定する必要があります。

このタスクではライセンスレベルを設定します。設定された変更を有効にする前にリロードが必要です。このタスクは、次の目的で使用できます。

- 現在のライセンスを変更する。
- 別のライセンスを追加する。たとえば、現在 Network Advantage を使用している場合、対応する Digital Networking Architecture（DNA）Advantage ライセンスで使用可能な機能も使用することができます。
- ライセンスを削除する。

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 モードを有効にします。 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	**license boot level** *license_level* 例： `Device(config)# license boot level network-advantage add-on dna-advantage`	製品インスタンスで設定されたライセンスをアクティブにします。この例では、DNA Advantage ライセンスはリロード後に製品インスタンスでアクティブ化されます。
ステップ 4	**exit** 例： `Device(config)# exit`	特権 EXEC モードに戻ります。
ステップ 5	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	構成ファイルへの変更を保存します。
ステップ 6	**show version** 例： `Device# show version` `<output truncated>` `Technology Package License Information:` `Technology-package Technology-package Current Type Next reboot` `network-advantage Smart License network-advantage Subscription Smart License dna-advantage` `<output truncated>`	現在設定されているライセンスの情報と、リロード後に適用可能なライセンスを表示します。
ステップ 7	**reload** 例： `Device# reload`	デバイスがリロードされます。

次のタスク

ライセンスレベルを設定すると、変更はリロード後に有効になります。レポートが必要かどうかを確認するには、システムメッセージを待つか、show コマンドを使用してポリシーを参照します。

- レポートが必要であることを示すシステムメッセージ：`%SMART_LIC-6-REPORTING_REQUIRED: A Usage report acknowledgment will be required in [dec] days.`

  [dec] は、レポート要件を満たすために残された時間（日数）です。

- show コマンドを使用する場合は、**show license status** 特権 EXEC コマンドの出力を参照し、[Next ACK deadline] フィールドを確認します。これは、この日付までに RUM レポートを送信して ACK をインストールする必要があることを意味します。

RUM レポートを送信するために使用可能な方法は、実装するトポロジによって異なります。このガイドの「*Smart Licensing Using Policy*」の章のポリシーを使用したスマートライセンシングの設定方法：トポロジ別のワークフロー （164 ページ）セクションで、該当するトポロジのワークフローを参照してください。

HSECK9 ライセンス用の SLAC のインストール

このセクションでは、HSECK9 ライセンス用の SLAC をインストールする各種方法について説明します。各方法は、Smart Licensing Using Policy 環境の特定のトポロジに対応します。

サポートされているすべてのトポロジの詳細については、このガイドの「Smart Licensing Using Policy」章のサポートされるトポロジ （147 ページ）セクションを参照してください。

（注） HSECK9 ライセンスを使用する場合に実装できない唯一のトポロジは、「コントローラを介して CSSM に接続」です。ここで、「コントローラ」は Cisco DNA Center を指します。Cisco DNA Center GUI には、HSECK9 をサポートする Cisco Catalyst スイッチの SLAC を生成するオプションはありません。

SLAC のインストール：CSSM に直接接続

このタスクでは、デバイス（製品インスタンス）が CSSM に直接接続されている場合に、SLAC を要求してインストールする方法を示します。

始める前に

- デバイスが HSECK9 ライセンスをサポートしていることを確認します。サポートされているプラットフォームとリリース （111 ページ）を参照してください。
- CSSM の該当するスマートアカウントおよびバーチャルアカウントに必要な数の HSECK9 ライセンスがあることを確認します。
- *CSSM* に直接接続トポロジのステップ 1 ー 3 が完了していることを確認します。トポロジのワークフロー：CSSM に直接接続 （168 ページ）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	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。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	**license smart authorization request{add | replace}***feature_name*{**all | local**} 例： `Device# license smart authorization request add hseck9 local`	• **license smart authorization request**：このオプションは、CSSM、CSLU またはSSMオンプレミスからSLACを要求します。SLACが返され、製品インスタンスに自動的にインストールされます。 既存のSLACに追加するのか置換するのかを指定します。 • **add**：要求されたライセンスを既存のSLACに追加します。新しい承認コードには、既存のSLACのすべてのライセンスと要求されたライセンスが含まれます。 • **replace**：既存の SLAC を置き換えます。新しいSLACには、要求されたライセンスのみが含まれます。既存のSLACのすべてのライセンスが返却されます。このキーワードを入力すると、製品インスタンスはこれらの既存のライセンスが使用中かどうかを確認します。使用中の場合は、対応する機能を最初に無効にするようにエラーメッセージが表示されます。 *feature_name*：SLAC の追加または置換を要求するライセンスの名前を入力します。「hseck9」と入力して、HSECK9 ライセンスの SLAC を要求してインストールします。 次のいずれかのオプションを入力して、デバイスを指定します。 • **all**：高可用性設定およびスタック設定のすべてのデバイスの承認コードを取得します。

	コマンドまたはアクション	目的
		（注） SLAC がすでにインストールされている既存のスタックに（SLAC がインストールされていない）デバイスを追加した場合は、replace および all オプションを使用します。これにより、スタック内のすべてのデバイスの SLAC が要求されます。特定のメンバーの SLAC を要求することはできません。選択肢はアクティブまたはスタック全体のみです。 • **local**：高可用性設定およびスタック設定のアクティブなデバイスの承認コードを取得します。これがデフォルトのオプションです。
ステップ 3	（任意） **license smart sync{all | local}** 例： `Device# license smart sync local`	CSSM、CSLU または SSM オンプレミスと製品インスタンスの同期がトリガーされ、保留中のデータが送受信されます。 この手順はオプションであり、製品インスタンスが CSSM、CSLU または SSM オンプレミスに接続され、製品インスタンスが通信を開始するシナリオにのみ適用されます。対応するトポロジは、CSSM に直接接続、CSLU を介して CSSM に接続（製品インスタンス開始）、および SSM オンプレミス展開（製品インスタンス開始型通信）です。 このコマンドは、手動で同期をトリガーし、SLAC インストールプロセスを完了します。それ以外の場合、製品インスタンスが次回 CSLU または SSM オンプレ

	コマンドまたはアクション	目的
		ミスに接続するときに、SLAC が製品インスタンスに適用されます。

次のタスク

SLAC のインストール後に必要なタスク （129 ページ）

SLAC のインストール：CSSM への接続なし、CSLU なし

このタスクでは、デバイス（製品インスタンス）がネットワーク外のデバイスとオンラインで通信できない、外部との接続姓がないネットワークにSLACを要求してインストールする方法を示します。

このタスクは2つの部分で構成されます。最初の部分（最初のステップ）では、CSCMから各HSECK9ライセンスのSLACファイルを生成してダウンロードする必要があります。インターネットおよびCSSM Web UIに接続できるワークステーションが必要です。ステップ2以降は、ダウンロードしたSLACファイルを製品インスタンスにインポートするために設定する必要があるコマンドです。

始める前に

- デバイスが HSECK9 ライセンスをサポートしていることを確認します。サポートされているプラットフォームとリリース （111 ページ）を参照してください。
- CSSM の該当するスマートアカウントおよびバーチャルアカウントに必要な数の HSECK9 ライセンスがあることを確認します。
- *CSSM* への接続なし、*CSLU* なしトポロジのステップ 1 が完了していることを確認します。トポロジのワークフロー：CSSM への接続なし、CSLU なし （175 ページ）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	CSSMからのSLACの生成とファイルへのダウンロード （249 ページ）	このタスクは、CSSM Web UI で実行します。
ステップ 2	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。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 3	**copy** *source* **bootflash:***file-name* 例： `Device# copy tftp://10.8.0.6/bootflash:example.txt`	（任意）ファイルをソースの場所またはディレクトリから製品インスタンスのフラッシュメモリにコピーします。また、リモートの場所からファイルを直接イン

	コマンドまたはアクション	目的
		ポートし、製品インスタンスにインストールすることもできます（次の手順）。 • コピー元：これはファイルのコピー元の場所です。コピー元は、ローカルまたはリモートのいずれかです。 • **bootflash:**：これはブートフラッシュメモリの場合の宛先です。
ステップ 4	**license smart import** *filepath_file-name* 例： `Device# license smart import bootflash:example.txt`	ファイルを製品インスタンスにインポートしてインストールします。*filepath_filename* には、場所（ファイル名を含む）を指定します。インストール後、インストールしたファイルのタイプを示すシステムメッセージが表示されます。 （注） 複数の製品インスタンスに SLAC をインストールする場合（スタック設定など）、UDI ごとに個別の .txt SLAC ファイルをダウンロードしてください。一度に 1 つのファイルをインポートしてインストールします。

次のタスク

SLAC のインストール後に必要なタスク （129 ページ）

SLAC のインストール：CSLU を介した CSSM への接続（製品インスタンス開始）

このタスクでは、デバイス（製品インスタンス）が CSLU を介して CSSM に接続され、製品インスタンスが通信を開始する場合、つまり製品インスタンスが必要な情報を CSLU にプッシュするように設定されている場合に、SLAC を要求してインストールする方法を示します。

始める前に

- デバイスが HSECK9 ライセンスをサポートしていることを確認します。サポートされているプラットフォームとリリース （111 ページ）を参照してください。
- CSSM の該当するスマートアカウントおよびバーチャルアカウントに必要な数の HSECK9 ライセンスがあることを確認します。

- CSLU を介した CSSM への接続（製品インスタンス開始型通信）トポロジのステップ 1 ー 3 が完了していることを確認します。トポロジのワークフロー：CSLU を介して CSSM に接続 （165 ページ）→製品インスタンス開始型通信の場合のタスク （165 ページ）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	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。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	**license smart authorization request** {**add** | **replace**} *feature_name* {**all** | **local**} 例： `Device# license smart authorization request add hseck9 local`	• **license smart authorization request**：このオプションは、CSSM、CSLU またはSSM オンプレミスから SLAC を要求します。SLAC が返され、製品インスタンスに自動的にインストールされます。 既存の SLAC に追加するのか置換するのかを指定します。 • **add**：要求されたライセンスを既存の SLAC に追加します。新しい承認コードには、既存の SLAC のすべてのライセンスと要求されたライセンスが含まれます。 • **replace**：既存の SLAC を置き換えます。新しい SLAC には、要求されたライセンスのみが含まれます。既存の SLAC のすべてのライセンスが返却されます。このキーワードを入力すると、製品インスタンスはこれらの既存のライセンスが使用中かどうかを確認します。使用中の場合は、対応する機能を最初に無効にするようにエラーメッセージが表示されます。 *feature_name*：SLAC の追加または置換を要求するライセンスの名前を入力します。「hseck9」と入力し

	コマンドまたはアクション	目的
		て、HSECK9 ライセンスの SLAC を要求してインストールします。 次のいずれかのオプションを入力して、デバイスを指定します。 • **all**：高可用性設定およびスタック設定のすべてのデバイスの承認コードを取得します。 （注） SLAC がすでにインストールされている既存のスタックに（SLAC がインストールされていない）デバイスを追加した場合は、replace および all オプションを使用します。これにより、スタック内のすべてのデバイスの SLAC が要求されます。特定のメンバーの SLAC を要求することはできません。選択肢はアクティブまたはスタック全体のみです。 • **local**：高可用性設定およびスタック設定のアクティブなデバイスの承認コードを取得します。これがデフォルトのオプションです。
ステップ 3	（任意） **license smart sync{all|local}** 例： `Device# license smart sync local`	CSSM、CSLU または SSM オンプレミスと製品インスタンスの同期がトリガーされ、保留中のデータが送受信されます。 この手順はオプションであり、製品インスタンスが CSSM、CSLU または SSM オンプレミスに接続され、製品インスタンスが通信を開始するシナリオにのみ適用されます。対応するトポロジは、CSSM に直接接続、CSLU を介して

	コマンドまたはアクション	目的
		CSSM に接続（製品インスタンス開始）、および SSM オンプレミス展開（製品インスタンス開始型通信）です。 このコマンドは、手動で同期をトリガーし、SLAC インストールプロセスを完了します。それ以外の場合、製品インスタンスが次回 CSLU または SSM オンプレミスに接続するときに、SLAC が製品インスタンスに適用されます。

次のタスク

SLAC のインストール後に必要なタスク （129 ページ）

SLAC のインストール：CSLU を介した CSSM への接続（CSLU 開始）

このタスクでは、デバイス（製品インスタンス）が CSLU を介して CSSM に接続され、CSLU が通信を開始する場合、つまり CSLU が必要な情報を製品インスタンスからプルするように設定されている場合に、SLAC を要求してインストールする方法を示します。

このタスクでは、製品インスタンスの特定のコマンド、CSSM Web UI の特定のタスク、および CSLU インターフェイスの特定のタスクを設定する必要があります。

始める前に

- デバイスが HSECK9 ライセンスをサポートしていることを確認します。サポートされているプラットフォームとリリース （111 ページ）を参照してください。
- CSSM の該当するスマートアカウントおよびバーチャルアカウントに必要な数の HSECK9 ライセンスがあることを確認します。
- CSLU を介した CSSM への接続（製品インスタンス開始型通信）トポロジのステップ 1 ー 3 が完了していることを確認します。トポロジのワークフロー：CSLU を介して CSSM に接続 （165 ページ）→CSLU 開始型通信の場合のタスク （167 ページ）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	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。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	**license smart authorization request** {**add** | **replace**} *feature_name* {**all** | **local**}	• **license smart authorization request**：このオプションは、CSSM、CSLU

	コマンドまたはアクション	目的
	例： `Device# license smart authorization request add hseck9 local`	またはSSMオンプレミスからSLACを要求します。SLACが返され、製品インスタンスに自動的にインストールされます。 既存のSLACに追加するのか置換するのかを指定します。 • **add**：要求されたライセンスを既存のSLACに追加します。新しい承認コードには、既存のSLACのすべてのライセンスと要求されたライセンスが含まれます。 • **replace**：既存の SLAC を置き換えます。新しいSLACには、要求されたライセンスのみが含まれます。既存のSLACのすべてのライセンスが返却されます。このキーワードを入力すると、製品インスタンスはこれらの既存のライセンスが使用中かどうかを確認します。使用中の場合は、対応する機能を最初に無効にするようにエラーメッセージが表示されます。 *feature_name*：SLAC の追加または置換を要求するライセンスの名前を入力します。「hseck9」と入力して、HSECK9 ライセンスの SLAC を要求してインストールします。 次のいずれかのオプションを入力して、デバイスを指定します。 • **all**：高可用性設定およびスタック設定のすべてのデバイスの承認コードを取得します。

	コマンドまたはアクション	目的
		（注） SLAC がすでにインストールされている既存のスタックに（SLAC がインストールされていない）デバイスを追加した場合は、replace および all オプションを使用します。これにより、スタック内のすべてのデバイスの SLAC が要求されます。特定のメンバーの SLAC を要求することはできません。選択肢はアクティブまたはスタック全体のみです。 • **local**：高可用性設定およびスタック設定のアクティブなデバイスの承認コードを取得します。これがデフォルトのオプションです。
ステップ 3	1 つ以上の製品インスタンスの SLAC の要求（CSLU インターフェイス） （221 ページ）	このタスクは、CSLU インターフェイスで実行します。
ステップ 4	CSSM からの SLAC の生成とファイルへのダウンロード （249 ページ）	このタスクは、CSSM Web UI で実行します。
ステップ 5	CSSM からのインポート（CSLU インターフェイス） （216 ページ）	このタスクは、CSLU インターフェイスで実行します。完了したら、CSLU が次に更新を実行するときに、アップロードされたコードが製品インスタンスに適用されます。

次のタスク

SLAC のインストール後に必要なタスク （129 ページ）

SLAC のインストール：SSM オンプレミス展開（製品インスタンス開始）

このタスクでは、デバイス（製品インスタンス）が SSM オンプレミスに接続され、製品インスタンスが通信を開始する場合、つまり製品インスタンスが必要な情報を SSM オンプレミスにプッシュするように設定されている場合に、SLAC を要求してインストールする方法を示します。

ここでは、最初に SSM オンプレミスで要求ファイルを作成し、CSSM Web UI で要求をアップロードし、SLAC を生成して、SLAC を SSM オンプレミスサーバーにインポートします。最後に、SLAC を要求してインストールするように製品インスタンスのコマンドを設定します。

始める前に

- デバイスが HSECK9 ライセンスをサポートしていることを確認します。サポートされているプラットフォームとリリース （111 ページ）を参照してください。
- CSSM の該当するスマートアカウントおよびバーチャルアカウントに必要な数の HSECK9 ライセンスがあることを確認します。
- SSM オンプレミス展開（製品インスタンス開始）トポロジのステップ 1 ー 3 c. を完了していることを確認します。トポロジのワークフロー：SSM オンプレミス展開 （176 ページ）→製品インスタンス開始型通信の場合のタスク （176 ページ）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	承認コード要求の送信（SSM オンプレミス UI） （243 ページ）	このタスクは、SSM オンプレミス UI で実行します。
ステップ 2	CSSM からの SLAC の生成とファイルへのダウンロード （249 ページ）	このタスクは、CSSM Web UI で実行します。
ステップ 3	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。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 4	**license smart authorization request** {**add** | **replace**} *feature_name* {**all** | **local**} 例： `Device# license smart authorization request add hseck9 local`	• **license smart authorization request**：このオプションは、CSSM、CSLU または SSM オンプレミスから SLAC を要求します。SLAC が返され、製品インスタンスに自動的にインストールされます。 既存の SLAC に追加するのか置換するのかを指定します。 • **add**：要求されたライセンスを既存の SLAC に追加します。新

	コマンドまたはアクション	目的
		しい承認コードには、既存の SLAC のすべてのライセンスと要求されたライセンスが含まれます。 • **replace**：既存の SLAC を置き換えます。新しい SLAC には、要求されたライセンスのみが含まれます。既存の SLAC のすべてのライセンスが返却されます。このキーワードを入力すると、製品インスタンスはこれらの既存のライセンスが使用中かどうかを確認します。使用中の場合は、対応する機能を最初に無効にするようにエラーメッセージが表示されます。 *feature_name*：SLAC の追加または置換を要求するライセンスの名前を入力します。「hseck9」と入力して、HSECK9 ライセンスの SLAC を要求してインストールします。 次のいずれかのオプションを入力して、デバイスを指定します。 • **all**：高可用性設定およびスタック設定のすべてのデバイスの承認コードを取得します。

	コマンドまたはアクション	目的
		（注） SLAC がすでにインストールされている既存のスタックに（SLAC がインストールされていない）デバイスを追加した場合は、replace および all オプションを使用します。これにより、スタック内のすべてのデバイスの SLAC が要求されます。特定のメンバーの SLAC を要求することはできません。選択肢はアクティブまたはスタック全体のみです。 • **local**：高可用性設定およびスタック設定のアクティブなデバイスの承認コードを取得します。これがデフォルトのオプションです。
ステップ 5	（任意） **license smart sync{all|local}** 例： `Device# license smart sync local`	CSSM、CSLU または SSM オンプレミスと製品インスタンスの同期がトリガーされ、保留中のデータが送受信されます。 この手順はオプションであり、製品インスタンスが CSSM、CSLU または SSM オンプレミスに接続され、製品インスタンスが通信を開始するシナリオにのみ適用されます。対応するトポロジは、CSSM に直接接続、CSLU を介して CSSM に接続（製品インスタンス開始）、および SSM オンプレミス展開（製品インスタンス開始型通信）です。 このコマンドは、手動で同期をトリガーし、SLAC インストールプロセスを完了します。それ以外の場合、製品インスタンスが次回 CSLU または SSM オンプレ

	コマンドまたはアクション	目的
		ミスに接続するときに、SLAC が製品インスタンスに適用されます。

次のタスク

SLAC のインストール後に必要なタスク （129 ページ）

SLAC のインストール：SSM オンプレミス展開（SSM オンプレミス開始）

このタスクでは、デバイス（製品インスタンス）が SSM オンプレミスに接続され、SSM オンプレミスが通信を開始する場合（つまり、SSM オンプレミスが製品インスタンスから必要な情報をプルするように設定されている場合）に、SLAC を要求してインストールする方法を示します。

ここでは、SSM オンプレミスで要求ファイルを作成し、CSSM Web UI で要求をアップロードし、SLAC を生成して、SSM オンプレミスサーバーにインポートします。最後に、SSM オンプレミスを製品インスタンスと同期します。

始める前に

- デバイスが HSECK9 ライセンスをサポートしていることを確認します。サポートされているプラットフォームとリリース （111 ページ）を参照してください。
- CSSM の該当するスマートアカウントおよびバーチャルアカウントに必要な数の HSECK9 ライセンスがあることを確認します。
- SSM オンプレミス展開（製品インスタンス開始）トポロジのステップ 1 ー 3 a. を完了していることを確認します。トポロジのワークフロー：SSM オンプレミス展開 （176 ページ）→SSM オンプレミスインスタンス開始型通信の場合のタスク （179 ページ）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	承認コード要求の送信（SSM オンプレミス UI） （243 ページ）。	このタスクは、SSM オンプレミス UI で実行します。
ステップ 2	SSM オンプレミス UI で、[Reports] > [Synchronisation pull schedule with the devices] > [Synchronize now with the device] に移動します。	この手順は任意です。コードのインポート直後に同期を行わない場合、SSM オンプレミスが次に更新を実行するときに、アップロードされたコードが製品インスタンスに適用されます。

次のタスク

SLAC のインストール後に必要なタスク （129 ページ）

SLAC のインストール後に必要なタスク

このタスクでは、SLAC のインストール後に実行する必要があるアクティビティを示します。ここでの情報は、SLAC のインストール方法すべてに適用されます。

手順

ステップ 1 SLAC のインストールと HSECK9 ライセンスの使用を確認します。

- **show license authorization** 特権 EXEC コマンドの出力の承認ステータスが、Status: SMART AUTHORIZATION INSTALLED on <timestamp> と表示されていることを確認します。これは、SLAC がインストールされていることを意味します。複数の SLA Cを（高可用性またはスタック構成セットアップで）インストールした場合は、接続されているすべてのデバイスに上記のステータスが表示されていることを確認します。
- **show license summary** 特権 EXEC コマンドの出力で、使用状況ステータスとカウントに [NOT IN USE] と 0 が表示されていることを確認します。これは、HSECK9 ライセンスは使用可能ですが、まだ使用されていないことを意味します。
- SLAC のインストール後に、次のシステムメッセージが表示されます。

  `Error Message %SMART_LIC-6-AUTHORIZATION_INSTALL_SUCCESS: A new licensing authorization code was successfully installed on: [chars].` [chars] は、承認コードが正常にインストールされた UDI です。

  `%SMART_LIC-6-EXPORT_CONTROLLED: Usage of export controlled features is allowed for feature hseck9.`

例：

```
Device# show license authorization

Overall status:
  Active: PID:C9300X-24HX,SN:FOC2519L8R7
      Status: SMART AUTHORIZATION INSTALLED on Oct 29 17:45:28 2021 UTC
      Last Confirmation code: 6746c5b5
  Standby: PID:C9300X-48HXN,SN:FOC2524L39P
      Status: NOT INSTALLED
  Member: PID:C9300X-48HX,SN:FOC2516LC92
      Status: NOT INSTALLED

Authorizations:
  C9K HSEC (Cat9K HSEC):
    Description: HSEC Key for Export Compliance on Cat9K Series Switches
    Total available count: 1
    Enforcement type: EXPORT RESTRICTED
    Term information:
      Active: PID:C9300X-24HX,SN:FOC2519L8R7
        Authorization type: SMART AUTHORIZATION INSTALLED
        License type: PERPETUAL
          Term Count: 1
```

```
Purchased Licenses:
  No Purchase Information Available

Device# show license summary
License Usage:
  License                 Entitlement Tag               Count Status
  -----------------------------------------------------------------------------
  network-advantage       (C9300-24 Network Advan...)       1 IN USE
  dna-advantage           (C9300-24 DNA Advantage)          1 IN USE
  network-advantage       (C9300-48 Network Advan...)       2 IN USE
  dna-advantage           (C9300-48 DNA Advantage)          2 IN USE
  C9K HSEC                (Cat9K HSEC)                      0 NOT IN USE
```

ステップ 2 暗号化機能を設定します。

次の IPsec 設定は例を示すものにすぎません。機能の設定については、 Cisco IOS XE <リリース番号>（Catalyst 9300 スイッチ）『*Security Configuration Guide*』の「*Configuring IPsec*」の章を参照してください。

例：

```
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# int tu10
Device(config-if)# tunnel mode ipsec ipv4
Device(config-if)# end
```

ステップ 3 再度、HSECK9 ライセンスの使用状況を確認します。

暗号化機能を設定すると、**show license summary** 特権 EXEC コマンドの出力での使用状況とカウントが、[IN USE] と 1 に変わります。

（注） IN USE としてカウントされるのは、特定の時点で 1 つのライセンスのみです。

スタック構成のステップアップで複数のデバイスにSLACをインストールした場合でも、**show license summary** コマンド出力のライセンス使用カウントには 1 だけが表示されます。これは、特定の時点で 1 つの HSECK9 ライセンス（アクティブなライセンス）だけが使用されるためです。スイッチオーバーが発生すると、スタンバイの HSECK9 ライセンスが使用されます。スタンバイが新しくアクティブになっても、使用されているライセンスは 1 つであるため、使用カウントは 1 のままです。

例：

```
Device# show license summary
License Usage:
  License                 Entitlement Tag               Count Status
  -----------------------------------------------------------------------------
  network-advantage       (C9300-24 Network Advan...)       1 IN USE
  dna-advantage           (C9300-24 DNA Advantage)          1 IN USE
  network-advantage       (C9300-48 Network Advan...)       2 IN USE
  dna-advantage           (C9300-48 DNA Advantage)          2 IN USE
  hseck9                  (Cat9K HSEC)                      1 IN USE
```

ステップ 4 レポートが必要かどうかを確認します。RUM レポートを送信するために使用可能な方法は、実装するトポロジによって異なります。このガイドの「*Smart Licensing Using Policy*」の章のポリシーを使用したスマートライセンシングの設定方法：トポロジ別のワークフロー （164 ページ）セクションで、該当するトポロジのワークフローを参照してください。

レポートが必要かどうかを確認するには、システムメッセージを待つか、show コマンドを使用してポリシーを参照します。

- レポートが必要であることを示すシステムメッセージ：`%SMART_LIC-6-REPORTING_REQUIRED: A Usage report acknowledgement will be required in [dec] days.` [dec] は、レポート要件を満たすために残された時間（日数）です。
- show コマンドを使用する場合は、**show license status** 特権 EXEC コマンドの出力を参照し、[Next ACK deadline] フィールドを確認します。この場合、RUM レポートを送信し、この日付までに ACK がインストールされていることを確認する必要があります。

SLAC の返却

このタスクでは、SLAC を返却し、HSECK9 ライセンスを CSSM のライセンスプールに返却する方法を示します。このタスクは、すべてのトポロジで使用できます。

次の状況では、SLAC および HSECK9 ライセンスを返却することができます。

- デバイスで（HSECK9 ライセンスが必要な）暗号化機能を使用する必要がなくなった。
- 返品許可（RMA）のためにデバイスを返却するか、永久に使用を停止する。デバイスをシスコに返却する場合は、**licence smart factory reset** 特権 EXEC コマンドを設定する必要があります。これにより、承認コード、RUM レポートなどを含めて、すべてのライセンス情報（使用中のライセンスを除く）が製品インスタンスから削除されます。工場出荷時設定へのリセットを実行する前に、SLAC コードを返却します。また、製品インスタンスからライセンス情報を削除する前に、CSSM に RUM レポートを送信することが推奨されます。

始める前に

HSECK9 ライセンスを使用した暗号化機能を無効化または設定解除し、HSECK9 ライセンスのライセンス使用状況が [NOT IN USE] であることを確認します。

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 モードを有効にします。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 license summary** 例： `Device# show license summary` `License Usage:` `License Entitlement Tag Count Status`	（任意）ライセンスの使用状況の概要を表示します。この手順は、SLAC を返却する場合にのみ適用されます。 暗号化機能を無効にした後でも、HSECK9 ライセンスのステータスが [IN

	コマンドまたはアクション	目的
	``---------- network-advantage (C9300-24 Network Advan...) 1 IN USE dna-advantage (C9300-24 DNA Advantage) 1 IN USE network-advantage (C9300-48 Network Advan...) 2 IN USE dna-advantage (C9300-48 DNA Advantage) 2 IN USE C9K HSEC (Cat9K HSEC) 1 IN USE```	USE] と表示される場合は、次の手順を実行します。この例の場合を示します。 HSECK9 ライセンスのステータスが [NOT IN USE] と表示された場合は、ステップ 5 に進みます。
ステップ 3	**platform hsec-license-release** 例： ```Device# configure terminal Device(config)# platform hsec-license-release HSEC license is released Device(config)# exit```	（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、HSECK9 ライセンスを返却したら、特権 EXEC モードに戻ります。 HSECK9 ライセンスを使用する暗号化機能が無効または未設定で、ライセンスがまだ [IN USE] と表示されている場合、このコマンドにより HSECK9 ライセンスが強制的に [NOT IN USE] に変更されます。
ステップ 4	**show license summary** 例： ```Device# show license summary License Usage: License Entitlement Tag Count Status ---------- network-advantage (C9300-24 Network Advan...) 1 IN USE dna-advantage (C9300-24 DNA Advantage) 1 IN USE network-advantage (C9300-48 Network Advan...) 2 IN USE dna-advantage (C9300-48 DNA Advantage) 2 IN USE C9K HSEC (Cat9K HSEC) 0 NOT IN USE```	返却するライセンスのステータスが [NOT IN USE] であることを確認します。使用中の場合は、まず機能を無効にする必要があります。
ステップ 5	**license smart authorization return {all | local} {offline [*path*] | online}** 例： ```Device# license smart authorization return all online OR```	CSSM のライセンスプールに承認コードを返却します。このコマンドを入力すると、戻りコードが表示されます。 製品インスタンスを指定します。 • all：高可用性セットアップまたはスタック構成セットアップで接続され

	コマンドまたはアクション	目的
	Device# **license smart authorization return all offline** Enter this return code in Cisco Smart Software Manager portal: UDI: PID:C9300X-24HX,SN:FOC2519L8R7 Return code: Cr9JHx-L1x5Rj-ftwzg1-h9QZAU-LE5DT1-babWeL-FABPt9-Wr1Dn7-Rp7 OR Device# **license smart authorization return all offline bootflash:return-code.txt**	たすべての製品インスタンスに対してアクションを実行します。 • local：アクティブな製品インスタンスに対してアクションを実行します。これがデフォルトのオプションです。 CSSMに接続しているかどうかを指定します。 • CSSMに接続している場合、または製品インスタンス開始型通信のトポロジ（CSLU または SSM オンプレミス）を実装している場合は、**online** を入力します。コードは自動的に CSSM に返却され、確認が返されて製品インスタンスにインストールされます。このオプションを選択すると、戻りコードが自動的に CSSM に送信されます。 • CSSMに接続されていない場合、または CSLU 開始型通信または SSM オンプレミス開始型通信のトポロジを実装した場合は、**offline**[*filepath_filename*] を入力します。**offline** キーワードのみを入力する場合は、CLIに表示される戻りコードをコピーし、CSSMに入力します。戻りコードをファイルに保存する場合は、ファイルからコードをコピーし、CSSMに同じコードを入力できます。ファイル形式は、読み取り可能な任意の形式にすることができます（これはアップロードされません）。例：`Device# license smart authorization return local offline bootflash:return-code.txt` CSSM に戻りコードを入力するには、次のタスクを実行します。CSSM での SLAC 戻りコードの入力と製品インスタンスの削除 （255 ページ）

	コマンドまたはアクション	目的
ステップ 6	**show license authorization** 例： `Device# show license authorization` `License Authorizations` `==========` `Overall status:` `  Active:` `PID:C9300X-24HX,SN:FOC2519L8R7` `  Status: NOT INSTALLED` `  Last return code:` `Cr9JHx-L1x5Rj-ftwzg1-h9QZAU-LE5DT1-babWeL-FABPt9-Wr1Dn7-Rp7` `  Standby:` `PID:C9300X-48HXN,SN:FOC2524L39P` `  Status: NOT INSTALLED` `  Member:` `PID:C9300X-48HX,SN:FOC2516LC92` `  Status: NOT INSTALLED` `<output truncated>`	ライセンス情報を表示します。出力の `License Authorizations` ヘッダーを確認します。返却プロセスが正常に完了すると、`Last return code:` フィールドに戻りコードが表示されます。

使用可能なライセンスの機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

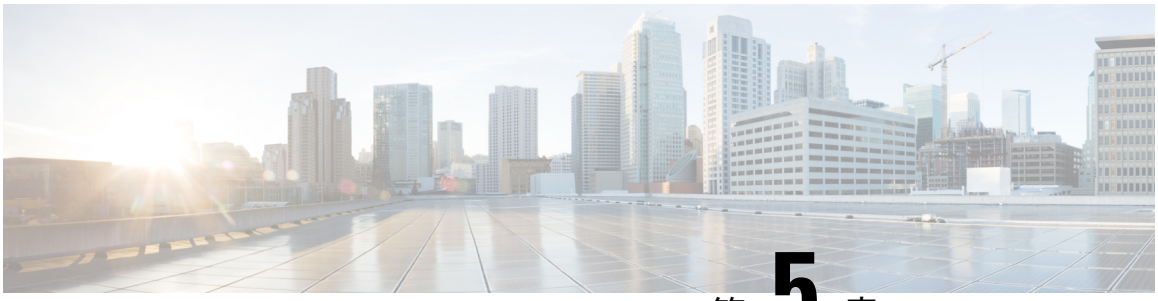
これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS XE Everest 16.5.1a	基本ライセンスとアドオンライセンス	Cisco Catalyst 9300 シリーズ スイッチで使用可能なソフトウェア機能は、基本ライセンスまたはアドオンライセンスレベルに分類されます。 基本ライセンスとアドオンライセンス（109ページ）および基本ライセンスとアドオンライセンスの設定（113ページ）を参照してください。

リリース	機能	機能情報
Cisco IOS XE Bengaluru 17.6.2	高セキュリティ（HSECK9）ライセンス	Cisco Catalyst 9300X シリーズ スイッチでの HSECK9 ライセンスのサポートを導入します。 HSECK9 ライセンスは、米国輸出管理法で制限されている暗号化機能の使用を許可する、輸出規制対象ライセンスです。制限付き暗号化機能を使用する場合は、HSECK9 ライセンスが必要です。高セキュリティライセンス（110ページ）およびHSECK9ライセンス用のSLACのインストール（115ページ）を参照してください。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator にアクセスするには、https://cfnng.cisco.com/にアクセスします。

http://www.cisco.com/go/cfn。



第 5 章

ポリシーを使用したスマートライセンシング

ポリシーを使用したスマートライセンシングの概要

ポリシーを使用したスマートライセンシングは、スマートライセンシングの拡張バージョンであり、ネットワークの運用を中断させないライセンスソリューションを提供するという主目的がありますが、むしろ、購入および使用しているハードウェアおよびソフトウェアライセンスを考慮してコンプライアンス関係を実現するライセンスソリューションを提供するという目的もあります。

Smart Licensing Using Policy は、Cisco IOS XE Amsterdam 17.3.2a 以降でサポートされます。

この拡張ライセンスモデルの主な利点は次のとおりです。

- シームレスな初日運用

  ライセンスを注文した後は、輸出規制または適用ライセンスを使用しない限り、キーの登録や生成などの準備手順は必要ありません。使用前に承認が必要なのはこれらのライセンスのみです。他のすべてのライセンスについては、製品機能をデバイスですぐに設定できます。

- Cisco IOS XE の一貫性

Cisco IOS XE ソフトウェアを実行するキャンパスおよび産業用イーサネットスイッチング、ルーティング、およびワイヤレスデバイスには、均一なライセンスエクスペリエンスがあります。

- 可視性と管理性

  使用中の情報を把握するためのツール、テレメトリ、製品タギング。

- コンプライアンスを維持するための柔軟な時系列レポート

  Cisco Smart Software Manager（CSSM）に直接または間接的に接続しているか、外部との接続性のないネットワークに接続しているかにかかわらず、簡単なレポートオプションを使用できます。

このドキュメントでは、Cisco Catalyst アクセス、コア、およびアグリゲーションスイッチでのポリシーを使用したスマートライセンシングの概念、設定、およびトラブルシューティングについて説明します。

シスコ ライセンスの詳細については、cisco.com/go/licensingguide を参照してください。

ポリシーを使用したスマートライセンシングに関する情報

このセクションでは、Smart Licensing Using Policy の実装に含めることができるコンポーネント、機能に関連する主要な概念、サポートされる製品、サポートされるすべてのトポロジの概要（機能を実装するさまざまな方法）、Smart Licensing Using Policy が他の機能とどのように連携するかについて説明します。

概要

ポリシーを使用したスマートライセンシングは、ライセンスのさまざまな側面をシームレスに体験できるソフトウェアライセンス管理ソリューションです。次に、この環境での操作の概要を示します。

- ライセンスの購入：既存のチャネルからライセンスを購入し、Cisco Smart Software Manager（CSSM）ポータルを使用して製品インスタンスとライセンスを表示します。

（注） ポリシーを使用したスマートライセンシングの実装を簡素化するには、新しいハードウェアまたはソフトウェアを注文する際にスマートアカウントとバーチャルアカウントの情報を提供します。これにより、シスコは製造時に該当するポリシーおよび承認コード（用語は以下のセクション概念 （142 ページ）で説明）をインストールできます。

- 使用：ほとんどのライセンスは適用（エンフォース）されません。つまり、ソフトウェアとそれに関連付けられているライセンスの使用を開始する前に、キーの登録や生成などのライセンス固有の操作を完了する必要はありません。輸出規制および適用されたライセンスのみ、使用前にシスコの承認が必要です。また、特定の製品のみが輸出規制ライセンスをサポートします。ライセンスの使用状況はタイムスタンプとともにデバイスに記録され、必要なワークフローは後日完了できます。
- ライセンスの使用状況をCSSMにレポート：ライセンス使用状況レポートには複数のオプションを使用できます。Cisco Smart Licensing Utility（CSLU）を使用し、使用状況情報をCSSM に直接報告し、コントローラ（Cisco DNA Center など）を使用し、Smart Software Managerオンプレミス（SSMオンプレミス）を展開して製品とライセンスをオンプレミスで管理できます。使用状況レポートはプレーンテキストの XML 形式です。リソース使用率測定レポートの例 （270 ページ）を参照してください。
- 調整：差分請求が適用される状況用（購入と消費を比較して差分がある場合）。

サポート対象製品

このセクションでは、本マニュアルの対象範囲に含まれる、ポリシーを使用したスマートライセンシングをサポートする Cisco IOS-XE 製品インスタンスについての情報を提供します。特に指定のない限り、製品シリーズのすべてのモデル（製品 ID または PID）がサポートされます。

表 *8*：サポートされる製品インスタンス：*Cisco Catalyst* アクセス、コア、およびアグリゲーションスイッチ

Cisco Catalyst アクセス、コア、およびアグリゲーションスイッチ	**サポートが導入されたバージョン**
Cisco Catalyst 9200 シリーズ スイッチ	Cisco IOS XE Amsterdam 17.3.2a
Cisco Catalyst 9300 シリーズ スイッチ	Cisco IOS XE Amsterdam 17.3.2a
Cisco Catalyst 9400 シリーズ スイッチ	Cisco IOS XE Amsterdam 17.3.2a
Cisco Catalyst 9500 シリーズ スイッチ	Cisco IOS XE Amsterdam 17.3.2a
Cisco Catalyst 9600 シリーズ スイッチ	Cisco IOS XE Amsterdam 17.3.2a

アーキテクチャ

ここでは、ポリシーを使用したスマートライセンシングの実装に含めることができるさまざまなコンポーネントについて説明します。

製品インスタンス

製品インスタンスとは、Unique Device Identifier（UDI）によって識別されるシスコ製品の単一インスタンスです。

製品インスタンスは、ライセンス使用状況（RUM レポート）を記録および報告し、期限切れのレポートや通信障害などに関するアラートとシステムメッセージを提供します。RUM レポートおよび使用状況データは、製品インスタンスに安全に保存されます。

このドキュメントでは、「製品インスタンス」という用語は、特に明記しない限り、サポートされているすべての物理および仮想製品インスタンスを指します。このドキュメントの範囲内にある製品インスタンスについては、サポート対象製品（139ページ）を参照してください。

CSLU

Cisco Smart License Utility（CSLU）は、集約ライセンスワークフローを提供する Windows ベースのレポートユーティリティです。このユーティリティが実行する主な機能は次のとおりです。

- ワークフローのトリガー方法に関するオプションを提供します。ワークフローは、CSLU や製品インスタンスによってトリガーできます。
- 製品インスタンスから使用状況レポートを収集し、その使用状況レポートを対応するスマートアカウントやバーチャルアカウントにアップロードします。オンラインでもオフライン（ファイルを使用）でも可能です。同様に、RUM レポート ACK をオンラインまたはオフラインで収集し、製品インスタンスに返送します。
- 承認コード要求を CSSM に送信し、CSSM から承認コードを受信します（該当する場合）。

CSLU は、次の方法で実装に含めることができます。

- CSSM に接続されているスタンドアロンツールとして CSLU を使用するための Windows アプリケーションをインストールします。
- CSSM から切断されているスタンドアロンツールとして CSLU を使用するための Windows アプリケーションをインストールします。このオプションを使用すると、必要な使用状況情報がファイルにダウンロードされ、CSSM にアップロードされます。これは、外部と接続していないネットワークに適しています。

CSSM

Cisco Smart Software Manager（CSSM）は、一元化された場所からすべてのシスコ ソフトウェア ライセンスを管理できるポータルです。CSSM は、現在の要件を管理し、将来のライセンス要件を計画するための使用傾向を確認するのに役立ちます。

CSSM Web UI には https://software.cisco.com でアクセスできます。[Smart Software Manager] で、[Manage licenses] リンクをクリックします。

このドキュメントのサポートされるトポロジ（147 ページ）では、CSSM に接続するさまざまな方法について説明します。

CSSM では、次のことができます。

- バーチャルアカウントを作成、管理、または表示する。
- バーチャルアカウント間または表示ライセンス間でライセンスを転送する。

- 製品インスタンスを転送、削除、または表示する。
- バーチャルアカウントに関するレポートを実行する。
- 電子メール通知の設定を変更する。
- 仮想アカウント情報を表示する。

コントローラ

複数の製品インスタンスを管理する管理アプリケーションまたはサービス。

Cisco Catalyst アクセス、コア、およびアグリゲーションスイッチでは、Cisco DNA Center がサポートされるコントローラです。コントローラ、コントローラをサポートする製品インスタンス、およびコントローラと製品インスタンスで必要な最小ソフトウェアバージョンに関する情報を次に示します。

表 *9 :* コントローラのサポート情報：*Cisco DNA Center*

Smart Licensing Using Policy へ移行するために必要な Cisco DNA Center の最小バージョン[3]	Cisco IOS XE に必要な最小バージョン[4]	サポート対象製品インスタンス
Cisco DNA Center リリース 2.2.2	Cisco IOS XE Amsterdam 17.3.2a	• Cisco Catalyst 9200 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ • Cisco Catalyst 9600 シリーズ スイッチ

[3] コントローラに必要な最小ソフトウェアバージョン。これは、特に明記されていない限り、後続のすべてのリリースでサポートが継続されることを意味します。

[4] 製品インスタンスに必要な最小ソフトウェアバージョン。これは、特に明記されていない限り、後続のすべてのリリースでサポートが継続されることを意味します。

Cisco DNA Center の詳細については、https://www.cisco.com/c/en/us/support/cloud-systems-management/dna-center/series.html でサポートページを参照してください。

SSM オンプレミス

Smart Software Manager オンプレミス（SSM オンプレミス）は、CSSM と連動するアセットマネージャです。これにより、CSSMに直接接続する代わりに、オンプレミスで製品とライセンスを管理できます。

SSM オンプレミスで Smart Licensing Using Policy を実装するために必要なソフトウェアバージョンについては、次を参照してください。

Smart Licensing Using Policy に必要な SSM オンプレミスの最小バージョン[5]	Cisco IOS XE に必要な最小バージョン[6]	サポート対象製品インスタンス
バージョン 8、リリース 202102	Cisco IOS XE Amsterdam 17.3.3	• Cisco Catalyst 9200 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ • Cisco Catalyst 9600 シリーズ スイッチ

[5] 必要な SSM オンプレミスの最小バージョンこれは、特に明記されていない限り、後続のすべてのリリースでサポートが継続されることを意味します。

[6] 製品インスタンスに必要な最小ソフトウェアバージョン。これは、特に明記されていない限り、後続のすべてのリリースでサポートが継続されることを意味します。

SSM オンプレミスの詳細については、ソフトウェアダウンロードページの Smart Software Manager On-Prem を参照してください。ドキュメントリンクを表示するには、.iso イメージにカーソルを合わせます。

概念

ここでは、ポリシーを使用したスマートライセンシングの主要な概念について説明します。

ライセンス執行（エンフォースメント）タイプ

所与のライセンスは、3 つの適用タイプのいずれかに属します。適用タイプは、ライセンスを使用する前に承認が必要かどうかを示します。

- 不適用または非適用

不適用ライセンスは、外部との接続がないネットワークで使用する前、または接続されたネットワークでの登録前に承認を必要としません。このようなライセンスの使用条件は、エンドユーザーライセンス契約（EULA）に基づきます。

Network Essentials、Network Advantage、Digital Network Architecture（DNA）Essentials、および DNA Advantage は、Cisco Catalyst アクセス、コア、およびアグリゲーションスイッチでサポートされる不適用ライセンスの例です。

- 適用

  この適用タイプに属するライセンスは、使用前に承認が必要です。必要な承認は承認コードの形式で行われ、対応する製品インスタンスにインストールする必要があります。

  適用ライセンスの例としては、シスコの産業用イーサネットスイッチで利用可能な Media Redundancy Protocol（MRP）クライアントライセンスがあります。

- 輸出規制

  この適用タイプに属するライセンスは米国の取引規制法によって輸出が制限されていて、これらのライセンスは使用前に承認が必要です。これらのライセンスの場合も、必要な承認コードは、対応する製品インスタンスにインストールする必要があります。シスコは、ハードウェア購入の際に発注がある場合、輸出規制ライセンスをプリインストールすることがあります。

  輸出規制ライセンスの例としては、シスコの特定のスイッチのみで使用可能な高速暗号化（HSECK9）ライセンスがあります。

ライセンス継続期間

これは、購入したライセンスが有効な期間を指します。所与のライセンスは、上記のいずれかの適用タイプに属し、次の期間有効です。

- 永久：このライセンスには使用期限日はありません。

  Network Essentials、Network Advantage、および HSECK9 は、永久ライセンスの例です。

- サブスクリプション：ライセンスは特定の日付まで有効です。

  DNA Essentials および DNA Advantage ライセンスは、サブスクリプション ライセンスの例です。

承認コード

スマートライセンシング承認コード（SLAC）は、輸出規制または適用（エンフォース）ライセンスの有効化および継続使用を可能にします。承認コードは製品インスタンスにインストールされます。使用しているライセンスに承認コードが必要な場合は、CSSM から要求できます。

SLAC を削除して CSSM ライセンスプールに戻すことができます。ただし、これを行うには、まずライセンスを使用する機能を無効にする必要があります。使用中の SLAC は返却できません。

表 *10 : SLAC* を必要とするライセンス、サポートされるプラットフォーム、およびリリース

適用タイプ	ライセンス	サポートされているプラットフォームとサポートされている導入リリース
輸出規制	HSECK9	Cisco IOS XE Bengaluru 17.6.2 以降の Cisco Catalyst 9300X シリーズ スイッチのみ。

HSECK9 ライセンスの詳細については、このガイドの「使用可能なライセンス」章の高セキュリティライセンス （110 ページ）セクションを参照してください。

SLR 承認コード

以前のライセンスモデルから Smart Licensing Using Policy にアップグレードする場合、固有の承認コードを使用する Specific License Reservation（SLR）を設定することができます。SLR 承認コードは、Smart Licensing Using Policy へのアップグレード後にサポートされます。

（注） 既存の SLR はアップグレード後に引き継がれますが、「予約」の概念が適用されないため、ポリシーを使用したスマートライセンシング環境で新しい SLR を要求することはできません。完全に外部との接続性がないネットワーク内にいる場合は、代わりに CSSM への接続なし、CSLU なしのトポロジが適用されます。

SLR 承認コードの処理方法の詳細については、アップグレード（158 ページ）を参照してください。SLR 承認コードを返す場合は、承認コードの返却（251 ページ）を参照してください。

ポリシー

ポリシーは、製品インスタンスに次のレポート手順を提供します。

- License usage report acknowledgement requirement (Reporting ACK required)：ライセンス使用状況レポートは RUM レポートと呼ばれ、確認応答は ACK と呼ばれます（「RUM レポートおよびレポート確認応答」を参照）。これは、この製品インスタンスのレポートに CSSM 確認応答が必要かどうかを指定する yes または no の値です。デフォルトポリシーは常に「yes」に設定されます。
- First report requirement (days)：最初のレポートは、ここで指定した期間内に送信される必要があります。

  この値がゼロの場合、最初のレポートは必要ありません。
- Reporting frequency (days)：後続のレポートは、ここで指定した期間内に送信される必要があります。

  ここの値がゼロの場合、使用状況が変更されない限り、以降のレポートは必要ありません。
- Report on change (days)：ライセンスの使用状況が変更された場合は、ここで指定した期間内にレポートが送信される必要があります。

ここの値がゼロの場合、使用状況の変更時のレポートは必要ありません。

ここの値がゼロでない場合は、変更を加えた後にレポートが必要です。次に示すすべてのシナリオは、製品インスタンスのライセンス使用状況における変更としてカウントされます。

- 消費されたライセンスの変更（別のライセンスへの変更やライセンスの追加または削除を含む）。
- ライセンスの消費なしから 1 つ以上のライセンスの消費への移行。
- 1 つ以上のライセンスの消費からライセンスの消費なしへの移行。

（注） 製品インスタンスがライセンスを使用していない場合、ポリシーのレポート要件（最初のレポート要件、レポート頻度、変更に関するレポート）のいずれかにゼロ以外の値が設定されていても、レポートは必要ありません。

ポリシー選択について

CSSMは、製品インスタンスに適用されるポリシーを決定します。特定の時点で使用されているポリシーは1つだけです。ポリシーとその値は、使用されているライセンスなど、さまざまな要因に基づいています。

`Cisco default`は、製品インスタンスで常に使用可能なデフォルトポリシーです。他のポリシーが適用されていない場合、製品インスタンスはこのデフォルトポリシーを適用します。次の表（表 11 : ポリシー：Cisco default （146 ページ））に、`Cisco default` ポリシー値を示します。

お客様はポリシーを設定することはできませんが、Cisco Global Licensing Operations チームに連絡して、カスタマイズされたポリシーを要求することができます。Support Case Manager に移動します。[OPEN NEW CASE] をクリックして、[Software Licensing] を選択します。ライセンスチームから、プロセスの開始や追加情報について連絡があります。カスタマイズされたポリシーは、CSSM のスマートアカウントを介して使用することもできます。

（注） 適用されているポリシー（使用中のポリシー）とそのレポート要件を確認するには、特権EXECモードで **show license all** コマンドを入力します。

表 *11 :* ポリシー：*Cisco default*

ポリシー：`Cisco default`	デフォルトポリシー値
Export (Perpetual/Subscription) （注） 適用タイプが「輸出規制」のライセンスにのみ適用されます。	Reporting ACK required：Yes First report requirement (days)：0 Reporting frequency (days)：0 Report on change (days)：0
Enforced (Perpetual/Subscription) （注） 適用タイプが「適用（エンフォース）」のライセンスにのみ適用されます。	Reporting ACK required：Yes First report requirement (days)：0 Reporting frequency (days)：0 Report on change (days)：0
Unenforced/Non-Export Perpetual[7]	Reporting ACK required：Yes First report requirement (days)：365 Reporting frequency (days)：0 Report on change (days)：90
Unenforced/Non-Export Subscription	Reporting ACK required：Yes First report requirement (days)：90 Reporting frequency (days)：90 Report on change (days)：90

[7] Unenforced/Non-Export Perpetual の場合：デフォルトポリシーの最初のレポート要件（365 日以内）は、ディストリビュータやパートナーからハードウェアやソフトウェアを購入した場合にのみ適用されます。

RUM レポートおよびレポート確認応答

リソース使用率測定レポート（RUM レポート）は、ポリシーで指定されたレポート要件を満たすために製品インスタンスが生成するライセンス使用状況レポートです。

確認応答（ACK）は CSSM からの応答であり、RUM レポートのステータスに関する情報を提供します。

製品インスタンスに適用されるポリシーによって、次のレポート要件が決まります。

- RUM レポートが CSSM に送信されるかどうか、およびこの要件を満たすために提供される最大日数。
- RUM レポートに CSSM からの確認応答（ACK）が必要かどうか。
- ライセンス消費の変化を報告するために提供される最大日数。

RUM レポートには、信頼コード要求や SLAC 要求などの他の要求が伴う場合があります。そのため、受信した RUM レポート ID に加えて、CSSM からの ACK には承認コード、信頼コード、およびポリシーファイルが含まれることがあります。

レポート方式、つまり CSSM への RUM レポートの送信方法は、実装するトポロジによって異なります。

信頼コード

製品インスタンスが RUM レポートに署名するために使用する、UDI に関連付けられた公開キー。これにより、改ざんが防止され、データの真正性が確保されます。

サポートされるトポロジ

このセクションでは、ポリシーを使用したスマートライセンシングを実装するさまざまな方法について説明します。各トポロジについて、付属の概要を参照してセットアップの動作設計を確認し、考慮事項と推奨事項（ある場合）を参照してください。

トポロジを選択した後

トポロジを選択した後、ポリシーを使用したスマートライセンシングの設定方法：トポロジ別のワークフロー （164 ページ） を参照してください。これらのワークフローは、新規展開のみに該当します。これらのワークフローにより、トポロジを実装する最も簡単で迅速な方法が実現します。

既存のライセンシングモデルから移行する場合は、ポリシーを使用したスマートライセンシングへの移行 （182 ページ） を参照してください。

追加の設定タスクを実行する場合（たとえば別のライセンスを設定する場合、アドオンライセンスを使用する場合、またはより短いレポート間隔を設定する場合）は、ポリシーを使用したスマートライセンシングのタスクライブラリ （210 ページ） を参照してください。続行する前に、入手可能な場合には「サポートされるトポロジ」を確認してください。

CSLU を介して CSSM に接続

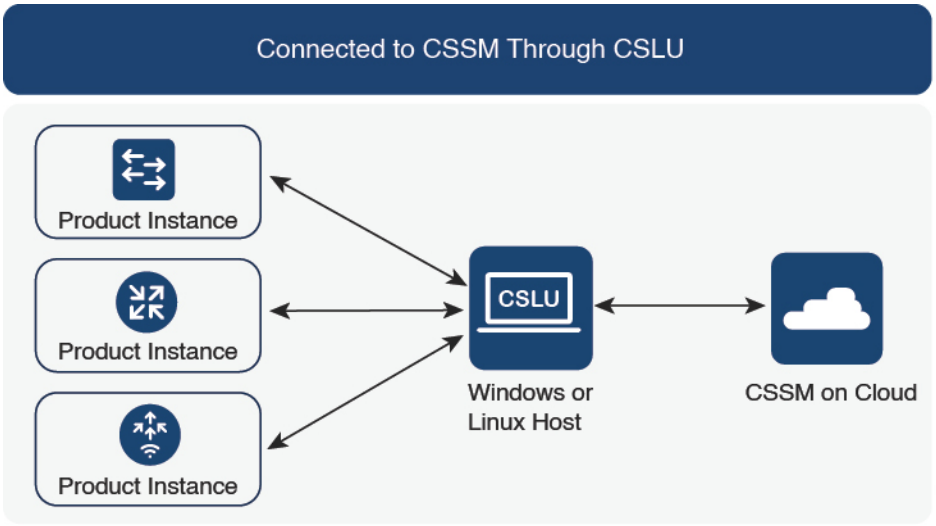
概要：

ここでは、ネットワーク内の製品インスタンスは CSLU に接続され、CSLU は CSSM との単一のインターフェイスポイントになります。製品インスタンスは、必要な情報を CSLU にプッシュするように設定できます。または、構成可能な頻度で製品インスタンスから必要な情報を取得するように CSLU を設定することもできます。

製品インスタンス開始型通信（プッシュ）：製品インスタンスは、CSLU の REST エンドポイントに接続することで、CSLU との通信を開始します。送信されるデータには、RUM レポート、および承認コード、信頼コードの要求が含まれます。必要な間隔で自動的に RUM レポートを CSLU に送信するように製品インスタンスを設定できます。これは、製品インスタンスのデフォルトの方法です。

CSLU 開始型通信（pull 型）：製品インスタンスからの情報の取得を開始するために、CSLU は YANG を使用した NETCONF、RESTCONF、gRPC のモデル、またはネイティブ REST API を使用して製品インスタンスに接続します。サポートされるワークフローには、RUM レポートの製品インスタンスからの受信と CSSM への送信、承認コードのインストール、信頼コードのインストール、およびポリシーの適用が含まれます。

図 *5:* トポロジ：*CSLU* を介して *CSSM* に接続



考慮事項または推奨事項：

ネットワークのセキュリティポリシーに応じて通信方法を選択します。

次の手順：

このトポロジを実装するには、トポロジのワークフロー：CSLU を介して CSSM に接続 （165 ページ）を参照してください。

CSSM に直接接続

概要：

このトポロジは、スマートライセンシングの以前のバージョンで使用でき、ポリシーを使用したスマートライセンシングで引き続きサポートされます。

ここでは、製品インスタンスから CSSM への直接かつ信頼できる接続を確立します。直接接続には、CSSM へのネットワーク到達可能性が要求されます。その後、製品インスタンスがメッセージを交換し、CSSM と通信するには、このトポロジで使用可能な転送オプションのいずれかを設定します（以下を参照）。最後に、信頼を確立するには、CSSM の対応するスマートアカウントとバーチャルアカウントからトークンを生成し、製品インスタンスにインストールする必要があります。

次の方法で CSSM と通信するように製品インスタンスを設定できます。

- スマート転送を使用して CSSM と通信する。

  スマート転送は、スマートライセンシング（JSON）メッセージが HTTPS メッセージ内に含まれ、製品インスタンスと CSSM の間で交換されることにより通信する転送方法です。次のスマート転送設定オプションを使用できます。

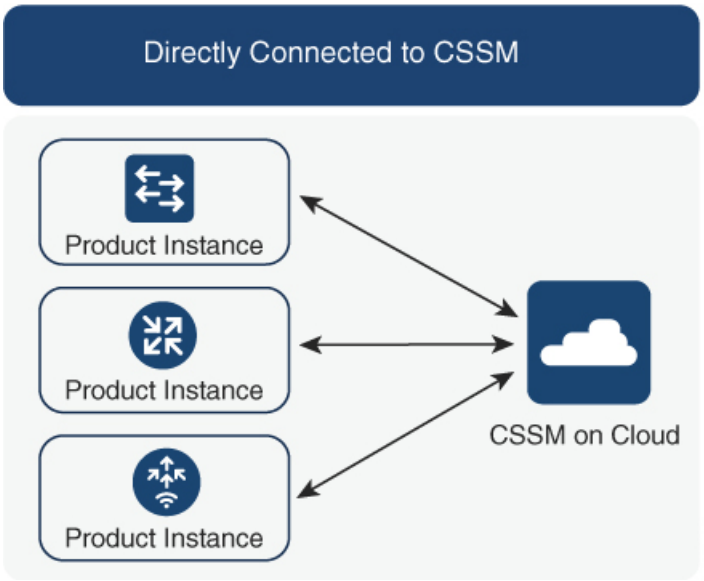
  - スマート転送：この方法では、製品インスタンスは特定のスマート転送ライセンスサーバー URL を使用します。これは、ワークフローのセクションに示すとおりに設定する必要があります。
  - HTTPS プロキシを介したスマート転送：この方法では、製品インスタンスはプロキシサーバーを使用してライセンスサーバーと通信し、最終的には CSSM と通信します。

- Call Home を使用して CSSM と通信する。

  Call Home を使用すると、E メールベースおよび Web ベースで重大なシステム イベントの通知を行えます。CSSM へのこの接続方法は、以前のスマートライセンシング環境で使用でき、ポリシーを使用したスマートライセンシングで引き続き使用できます。次の Call Home 設定オプションを使用できます。

  - ダイレクトクラウドアクセス：この方法では、製品インスタンスはインターネット経由で CSSM に使用状況情報を直接送信します。接続に追加のコンポーネントは必要ありません。
  - HTTPS プロキシを介したダイレクトクラウドアクセス：この方法では、製品インスタンスはインターネット経由でプロキシサーバー（Call Home Transport Gateway または市販のプロキシ（Apache など）のいずれか）を介して CSSM に使用状況情報を送信します。

図 *6 :* トポロジ：*CSSM* に直接接続



考慮事項または推奨事項：

CSSMに直接接続する場合は、スマート転送が推奨される転送方法です。この推奨事項は以下に適用されます。

- 新規展開。
- 以前のライセンスモデル。ポリシーを使用したスマートライセンシングへの移行後に設定を変更します。
- 現在 Call Home 転送方法を使用している登録済みライセンス。ポリシーを使用したスマートライセンシングへの移行後に設定を変更します。
- 以前のライセンスモデルの評価ライセンスや期限切れのライセンス。ポリシーを使用したスマートライセンシングへの移行後に設定を変更します。

移行後に設定を変更するには、トポロジのワークフロー：CSSM に直接接続 （168 ページ）の「製品インスタンスの設定」にある「接続方法と転送タイプの設定」のオプション1を参照してください。

次の手順：

このトポロジを実装するには、トポロジのワークフロー：CSSM に直接接続 （168 ページ）を参照してください。

コントローラを介して CSSM に接続

コントローラを使用して製品インスタンスを管理する場合、コントローラはCSSMに接続してCSSM とのすべての通信のインターフェイスとなります。Cisco Catalyst アクセス、コア、およびアグリゲーションスイッチのサポートされるコントローラは、Cisco DNA Center です。

概要

Cisco DNA Center がコントローラとして製品インスタンスを管理している場合、製品インスタンスはライセンスの使用状況を記録し、保存しますが、Cisco DNA Center が RUM レポートを取得し、CSSM に報告し、製品インスタンスにインストールするために ACK を返すために製品インスタンスとの通信を開始します。

Cisco DNA Center で管理する必要があるすべての製品インスタンスは、そのインベントリの一部である必要があり、サイトに割り当てる必要があります。Cisco DNA Center は NETCONF プロトコルを使用して設定をプロビジョニングし、製品インスタンスから必要な情報を取得します。したがって、これを容易にするために製品インスタンスでNETCONF を有効にする必要があります。

レポートの要件を満たすために、Cisco DNA Center は CSSM から該当するポリシーを取得し、次のレポートオプションを提供します。

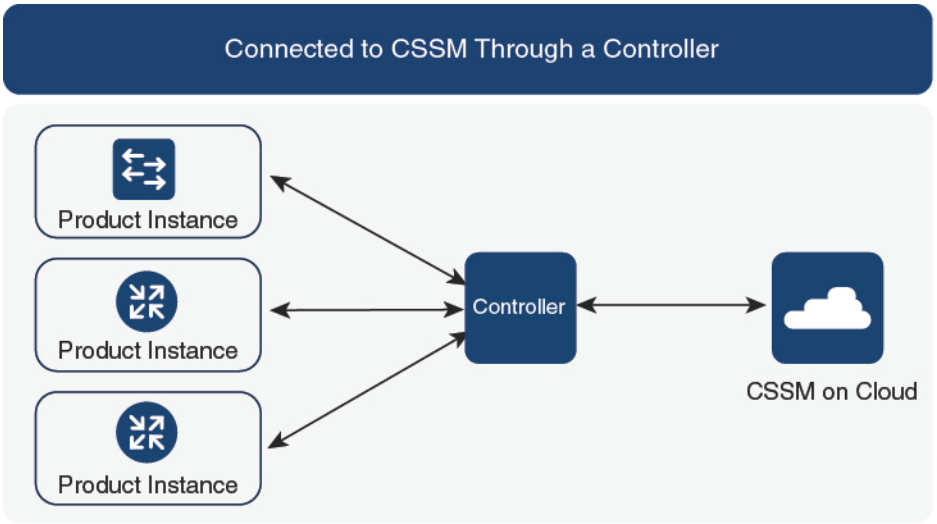
- Ad hoc reporting：必要に応じてアドホックレポートをトリガーできます。
- Scheduled reporting：ポリシーで指定されたレポート頻度に対応し、Cisco DNA Center によって自動的に処理されます。

（注） 製品インスタンスが定期レポートの対象となる前に、アドホックレポートを少なくとも1回実行する必要があります。

最初のアドホックレポートにより、Cisco DNA Center は、後続の RUM レポートをアップロードする必要があるスマートアカウントとバーチャルアカウントを決定できます。製品インスタンスのアドホックレポートが一度も実行されていない場合は、通知されます。

信頼コードは必要ありません。

図 *7:* トポロジ：コントローラを介して *CSSM* に接続



考慮事項または推奨事項：

これは、Cisco DNA Center を使用している場合に推奨されるトポロジです。

（注） 輸出規制ライセンスは、Cisco Catalyst アクセス、コア、およびアグリゲーションスイッチの特定のモデルでサポートされます（承認コード（143ページ）を参照）。HSECK9 ライセンスがサポートされている製品インスタンスを使用している場合は、Cisco DNA Center GUI に SLAC を生成するオプションが表示されないことに注意してください。

次の手順：

このトポロジを実装するには、トポロジのワークフロー：コントローラを介して CSSM に接続（170 ページ）を参照してください。

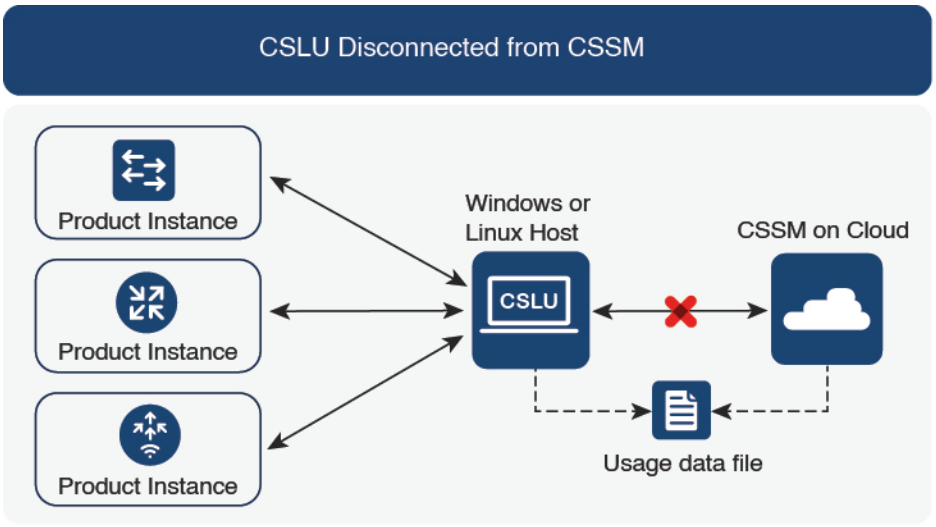
CSLU は CSSM から切断

概要：

ここでは、製品インスタンスが CSLU と通信し、製品インスタンス開始の通信または CSLU 開始の通信を実装するオプションがあります（CSLU を介して CSSM に接続のトポロジと同様）。CSLU と CSSM 間の通信のもう一方はオフラインです。CSLU には、CSSM から切断されたモードで動作するオプションがあります。

CSLU と CSSM 間の通信は、署名済みファイルの形式で送受信され、オフラインで保存された後、場合によっては CSLU または CSSM にアップロードまたはダウンロードされます。

図 *8*：トポロジ：*CSLU* は *CSSM* から切断



考慮事項または推奨事項：

なし。

次の手順：

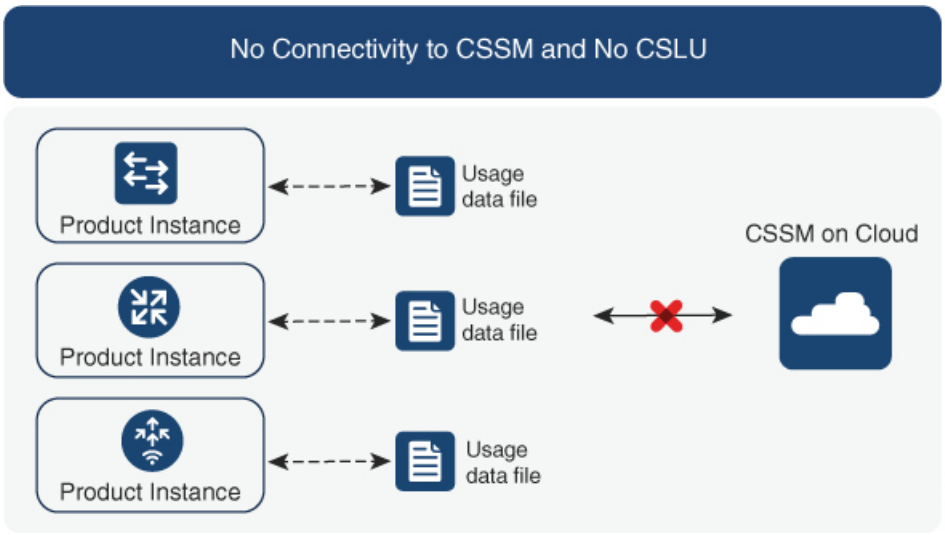
このトポロジを実装するには、トポロジのワークフロー：CSLU は CSSM から切断 （171 ページ）を参照してください。

CSSM への接続なし、CSLU なし

概要：

ここでは、製品インスタンスと CSSM は相互に切断され、他の中間ユーティリティまたはコンポーネントはありません。すべての通信は、ファイルのアップロードとダウンロードという形式です。

図 *9*：トポロジ：*CSSM* への接続なし、*CSLU* なし



考慮事項または推奨事項：

このトポロジは、製品インスタンスがネットワークの外部とオンラインで通信できない高セキュリティ展開に適しています。

次の手順：

このトポロジを実装するには、トポロジのワークフロー：CSSM への接続なし、CSLU なし（175 ページ）を参照してください。

SSM オンプレミス展開

概要：

SSM オンプレミスは、オンプレミスに展開される CSSM の拡張として機能するように設計されています。

ここでは、製品インスタンスが SSM オンプレミスに接続され、SSM オンプレミスが CSSM との単一のインターフェイスポイントになります。SSM オンプレミスの各インスタンスは、SSM オンプレミスのローカルアカウントに必須の登録と同期を通じて、CSSM 内のバーチャルアカウントを使用して CSSM に通知する必要があります。

製品インスタンスを管理するために SSM オンプレミスを展開する場合、SSM オンプレミスに必要な情報をプッシュするように製品インスタンスを設定できます。または、設定可能な頻度で製品インスタンスから必要な情報をプルするように SSM オンプレミスを設定することもできます。

- 製品インスタンス開始型通信（プッシュ）：製品インスタンスは SSM オンプレミスの REST エンドポイントを接続することで SSM オンプレミスの通信を開始します。送信され

るデータには、RUM レポート、および承認コード、信頼コード、ポリシーの要求が含まれます。

このモードでの製品インスタンスと SSM オンプレミス間の通信のオプション：

- 必要に応じて、CLI コマンドを使用して SSM オンプレミスに情報をプッシュします。
- スケジュールされた頻度で RUM レポートを SSM オンプレミスに自動的に送信するには、CLI コマンドを使用し、レポート間隔を設定します。

- SSM オンプレミス開始型通信（プル）：製品インスタンスからの情報の取得を開始するには、SSM オンプレミスで NETCONF、RESTCONF、およびネイティブの REST API オプションを使用して製品インスタンスを接続します。サポートされるワークフローには、RUM レポートの製品インスタンスからの受信と CSSM への送信、承認コードのインストール、信頼コードのインストール、およびポリシーの適用が含まれます。

  このモードでの製品インスタンスと SSM オンプレミス間の通信のオプション：

  - 必要に応じて（オンデマンドで）、1 つ以上の製品インスタンスから使用状況情報を収集します。
  - スケジュールされた頻度で 1 つ以上の製品インスタンスから使用状況情報を収集します。

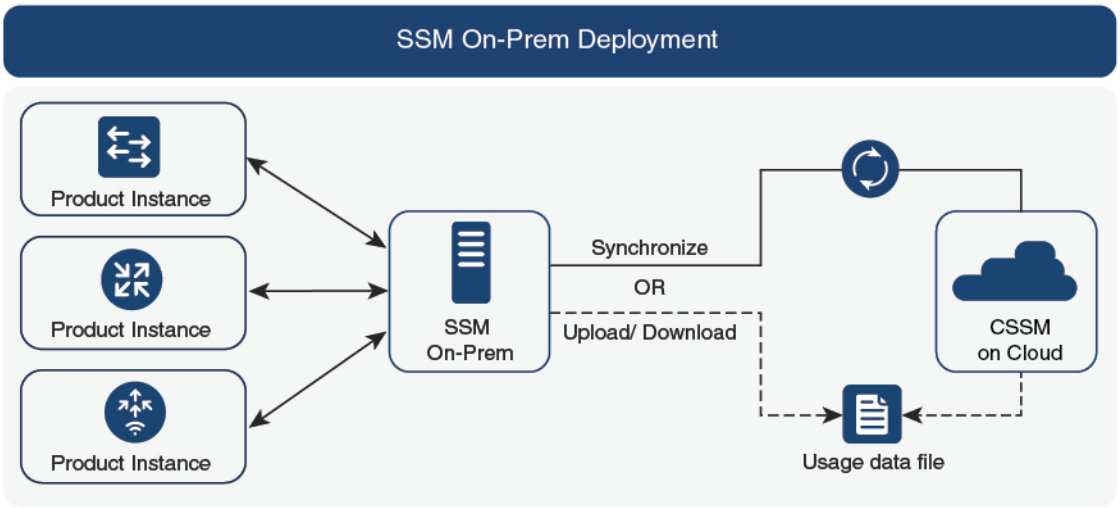
SSM オンプレミスでは、レポート間隔が製品インスタンスのデフォルトポリシーに設定されます。これは変更できますが、より頻繁に（より短い間隔で）レポートを作成するか、または使用可能な場合はカスタムポリシーをインストールできます。

SSM オンプレミスで使用状況が使用できるようになったら、同じ間隔で CSSM と同期して、製品インスタンス数、ライセンス数、およびライセンス使用状況情報が CSSM と SSM オンプレミスの両方と同じであることを確認します。SSM オンプレミスと CSSM 間の使用状況の同期オプション：プッシュとプルモードの場合：

- CSSM でアドホック同期を実行します（Cisco と同期されました）。
- 指定した時刻で CSSM との同期をスケジュールします。
- オフラインで保存されている指名済みファイルを通じて CSSM と通信し、場合によって SSM オンプレミスまたは CSSM からアップロードするか、またはダウンロードします。

（注） このトポロジでは、SSM オンプレミスと CSSM 間で 2 つの異なる同期が行われます。1 つは、ローカルアカウントと CSSM との同期です。この同期は、SSM オンプレミスインスタンスに CSSM を認識させるためであり、SSM オンプレミスの [Synchronization] ウィジェットを使用して実行します。2 番目は、CSSM に接続するか、またはファイルをダウンロードおよびアップロードすることのいずれかによるライセンスの使用状況の CSSM との同期です。ライセンスの使用状況を同期する前に、ローカルアカウントを同期する必要があります。

図 *10* : トポロジ : ***SSM*** オンプレミス展開



考慮事項または推奨事項 :

このトポロジは、次の状況に適しています。

- CSSM と直接通信せずにオンプレミスで製品インスタンスを管理する場合。
- 会社のポリシーにより、製品インスタンスでライセンスの使用状況をシスコ（CSSM）に直接報告できない場合。
- 製品インスタンスがエアギャップネットワーク内にあり、ネットワーク外にあるものとオンラインで通信できない場合。

Smart Licensing Using Policy のサポートとは別に、SSM オンプレミスのバージョン 8 の主な利点は次のとおりです。

- マルチテナント：1 つのテナントが 1 つのスマートアカウントとバーチャルアカウントのペアを構成します。SSM オンプレミスでは複数のペアを管理できます。ここでは、SSM オンプレミスに存在するローカルアカウントを作成します。CSSMのスマートアカウントとバーチャルアカウントのペアへの複数のローカルアカウントのロールアップ。詳細については、『Cisco Smart Software Manager On-Prem User Guide』の「About Accounts and Local Virtual Accounts」を参照してください。

（注） CSSM と SSM オンプレミスのインスタンス間の関係は、まだ 1 対 1 です。

- スケール：合計 300,000 の製品インスタンスをサポートします。
- 高可用性：2 台の SSM オンプレミスサーバーをアクティブ/スタンバイクラスタの形式で実行できます。詳細については、『Cisco Smart Software On-Prem Installation Guide』の

「Appendix 4 Managing a High Availability (HA) Cluster in Your System」を参照してください。

高可用性展開は SSM オンプレミスのコンソールでサポートされており、必要なコマンドの詳細については『Cisco Smart Software On-Prem Console Guide』で確認できます。

- CSSM へのオンライン接続とオフライン接続のオプション。

SSM オンプレミスの制限：

- SSM オンプレミスでは、ライセンス使用状況の同期を目的とする CSSM との通信へのプロキシの使用はサポートされていません。ただし、SSM オンプレミスではローカルアカウントの同期を目的とするプロキシの使用はサポートされてます。これは [Synchronization] ウィジェットを使用して実行します。
- SSM オンプレミス開始型通信は、ネットワークアドレス変換（NAT）設定の製品インスタンスではサポートされていません。製品インスタンス開始型通信を使用する必要があります。さらに、NAT 設定の製品インスタンスをサポートするために SSM オンプレミスを有効にする必要があります。詳細は、このトポロジのワークフローで提供されます。

次の手順：

このトポロジを実装するには、トポロジのワークフロー：SSM オンプレミス展開 （176 ページ）を参照してください。

SSM オンプレミスの既存のバージョンから移行する場合は、アップグレード関連のさまざまなアクティビティを実行する順序が重要です。「Smart Licensing Using Policy をサポートする SSM オンプレミスのバージョンへの移行 （208 ページ）」を参照してください。

他の機能との相互作用

ハイ アベイラビリティ

このセクションでは、ポリシーを使用したスマートライセンシングをサポートするソフトウェアバージョンを実行するときに、高可用性設定に適用される考慮事項について説明します。次の高可用性セットアップは、このドキュメントの範囲内です。

1つのアクティブ、1つのスタンバイ、および1つ以上のメンバーで構成されるデバイススタック

デュアル RP（ルートプロセッサ）セットアップ。1 つのシャーシに 2 つの RP がインストールされ、1 つはアクティブ、もう 1 つはスタンバイです。

デュアルシャーシ セットアップ[8]（固定またはモジュラ）。一方のシャーシにアクティブ、もう一方のシャーシにスタンバイがあります。

[8] Cisco Catalyst スイッチで使用可能な Cisco StackWise Virtual 機能が、このようなセットアップの例です。

モジュラシャーシでの、デュアルシャーシとデュアル RP のセットアップ[9]。ここでも 2 つのシャーシが関係し、1 つのシャーシにアクティブ RP、もう 1 つのシャーシにスタンバイ RP があります。デュアル RP とは、最小要件である 1 つのシャーシだけに追加のシャーシ内スタンバイ RP、または各シャーシにシャーシ内スタンバイ RP があることを指します。

高可用性セットアップでの信頼コード要件

必要な信頼コードの数は、UDIの数によって異なります。アクティブな製品インスタンスは、高可用性セットアップのすべてのデバイスに対する要求を送信し、ACK で返されるすべての信頼コードをインストールできます。

高可用性セットアップでのポリシー要件

高可用性セットアップにのみ適用されるポリシー要件はありません。スタンドアロン製品インスタンスの場合と同様に、高可用性セットアップにも1つのポリシーのみが存在し、これがアクティブになります。アクティブのポリシーは、セットアップのスタンバイまたはメンバーに適用されます。

高可用性セットアップでの製品インスタンス機能

このセクションでは、高可用性セットアップでの一般的な製品インスタンス機能と、新しいスタンバイまたはメンバーが既存の高可用性セットアップに追加された場合の製品インスタンスの動作について説明します。

承認コードと信頼コードの場合：アクティブな製品インスタンスは、スタンバイおよびメンバーの承認コードと信頼コードを（必要な場合に）要求し、インストールできます。

ポリシーの場合：アクティブな製品インスタンスがスタンバイと同期します。

レポートの場合：アクティブな製品インスタンスのみが使用状況を報告します。アクティブは、高可用性セットアップのすべてのデバイス（スタンバイまたはメンバーを適宜）の使用状況情報を報告します。

スケジュールされたレポートに加えて、次のイベントがレポートをトリガーします。

- スタンバイの追加または削除。RUM レポートには、追加または削除されたスタンバイに関する情報が含まれます。
- スタックマージおよびスタック分割イベントを含む、メンバーの追加または削除。RUM レポートには、追加または削除されたメンバーに関する情報が含まれます。
- スイッチオーバー。
- リロード。

上記のいずれかのイベントが発生すると、**show license status** 特権EXECコマンドの [Next report push] の日付が更新されます。ただし、レポートが製品インスタンスによって送信されるかどうかは、実装されたトポロジと関連するレポート方法で決まります。たとえば、製品インスタ

[9] Cisco Catalyst スイッチで使用可能なルートプロセッサ冗長性を備えたクアッドスーパーバイザが、このようなセットアップの例です。

ンスが切断されているトポロジ（[Transport Type] が [Off]）を実装した場合は、[Next report push] の日付が更新されても、製品インスタンスは RUM レポートを送信しません。

新規メンバーまたはスタンバイ追加の場合：

- CSLU に接続されている製品インスタンスは、それ以上のアクションを実行しません。
- CSSM に直接接続されている製品インスタンスは、信頼の同期を実行します。信頼の同期には、次のものが含まれます。

  スタンバイまたはメンバーに信頼コードがまだインストールされていない場合は、信頼コードのインストール。

  信頼コードがすでにインストールされている場合は、信頼の同期プロセスにより、新しいスタンバイまたはメンバーがアクティブと同じスマートアカウントおよびバーチャルアカウントにあることが保証されます。そうでない場合、新しいスタンバイまたはメンバーは、アクティブと同じスマートアカウントとバーチャルアカウントに移動されます。

  承認コード、ポリシー、および購入情報のインストール（該当する場合）

  現在の使用状況情報を含む RUM レポートの送信。

アップグレード

このセクションでは、ポリシーを使用したスマートライセンシングへのアップグレードまたは移行の処理方法について説明します。また、ポリシーを使用したスマートライセンシングが、以前のバージョンのスマートライセンシング、特定のライセンス予約（SLR）、使用権ライセンシング（RTU）を含む以前のライセンスモデルすべてを処理する方法、および以前のライセンシングモデルの評価ライセンスまたは期限切れライセンスがポリシーを使用したスマートライセンシング環境で処理される方法を具体的に説明します。

ポリシーを使用したスマートライセンシングに移行するには、ポリシーを使用したスマートライセンシングをサポートするソフトウェアバージョンにアップグレードする必要があります。アップグレードした後は、ポリシーを使用したスマートライセンシングが唯一のサポートされるライセンスモデルとなり、製品インスタンスはライセンスの変更なしで動作し続けます。ポリシーを使用したスマートライセンシングへの移行 （182 ページ） セクションでは、Cisco Catalyst アクセス、コア、およびアグリゲーションスイッチに適用される移行シナリオの詳細と例を示します。

デバイス先行の変換は、ポリシーを使用したスマートライセンシングへの移行ではサポートされていません。

アップグレード前に現在のライセンシングモデルを識別する

ポリシーを使用したスマートライセンシングにアップグレードする前に、製品インスタンスで有効な現在のライセンシングモデルを確認するには、特権 EXEC モードで **show license all** コマンドを入力します。このコマンドにより、RTU ライセンシングモデルを除くすべてのライセンシングモデルに関する情報が表示されます。**show license right-to-use** 特権 EXEC コマンドでは、ライセンシングモデルが RTU の場合にのみライセンス情報が表示されます。

アップグレードが既存ライセンスの適用タイプに与える影響

ポリシーを使用したスマートライセンシングをサポートするソフトウェアバージョンにアップグレードする場合、既存ライセンスの処理方法は、主に適用タイプによって決まります。

- アップグレード前に使用されていた不適用ライセンスは、アップグレード後も引き続き使用できます。これには、以前のすべてのライセンシングモデルのライセンスがすべて含まれます。
  - スマート ライセンス。
  - 特定のライセンス予約（SLR）。承認コードが付属しています。承認コードは、ポリシーを使用したスマートライセンシングへのアップグレード後も引き続き有効であり、既存のライセンスの使用を承認します。
  - 使用権（RTU）ライセンシング。
  - 上記のライセンシングモデルのいずれかの評価ライセンスまたは期限切れライセンス。
- アップグレード前に使用されていた適用ライセンスや輸出規制ライセンスは、必要な承認が存在する場合、アップグレード後も引き続き使用できます。

  輸出規制ライセンスは、Cisco IOS XE Bengaluru 17.6.2 以降の特定のモデルでのみサポートされています。それ以前の Cisco Catalyst アクセス、コア、およびアグリゲーションスイッチで使用可能な輸出規制ライセンスや適用ライセンスはありませんでした。

アップグレードが既存ライセンスのレポートに与える影響

既存ライセンス	ポリシーを使用したスマートライセンシングへの移行後のレポート要件
使用権（RTU）	使用されているライセンスによって異なります。 サポートされるトポロジの移行および展開後、**show license usage** コマンドの出力で `Next ACK deadline` フィールドを参照して、レポートが必要かどうか、およびいつ必要かを確認します。
特定のライセンス予約（SLR）	ライセンス消費に変更がある場合にのみ必要です。 既存の SLR 承認コードは、ポリシーを使用したスマートライセンシングへのアップグレード後に既存のライセンス消費を承認します。

既存ライセンス	ポリシーを使用したスマートライセンシングへの移行後のレポート要件
スマートライセンシング（登録済みライセンスと承認済みライセンス）：これらのライセンスのレポートは、ポリシーのレポート要件に基づいています。	ポリシーによって異なります。
評価ライセンスまたは期限切れライセンス	シスコのデフォルトポリシーのレポート要件に基づいています。

アップグレードが既存ライセンスの転送タイプに与える影響

既存の設定で転送タイプが設定されている場合、ポリシーを使用したスマートライセンシングへのアップグレード後も転送タイプが保持されます。

スマートライセンシングの以前のバージョンと比較した場合、ポリシーを使用したスマートライセンシングでは追加の転送タイプを使用できます。デフォルトの転送モードにも変更があります。次の表に、これがアップグレードに与える影響を示します。

アップグレード前の転送タイプ	アップグレード前のライセンスまたはライセンスの状態	アップグレード後の転送タイプ
デフォルト（callhome）	評価	cslu（ポリシーを使用したスマートライセンシングのデフォルト）
	SLR	off
	登録	callhome
smart	評価	off
	SLR	off
	登録	smart
N/A たとえば、既存のライセンスモデルが RTU の場合。	N/A たとえば、既存のライセンスモデルが RTU の場合。	cslu

アップグレードがトークン登録プロセスに与える影響

以前のバージョンのスマートライセンシングでは、CSSMへの登録と接続にトークンが使用されていました。ID トークンの登録は、ポリシーを使用したスマートライセンシングでは必要ありません。トークン生成機能はCSSMでも引き続き使用でき、製品インスタンスがCSSMに

直接接続されている場合に信頼を確立するために使用されます。「CSSMに直接接続」を参照してください。

ダウングレード

ダウングレードするには、製品インスタンスのソフトウェアバージョンをダウングレードする必要があります。このセクションでは、新規展開および既存の展開のダウングレードに関する情報を提供します（ポリシーを使用したスマートライセンシングにアップグレードした後にダウングレードする場合）。

新規展開のダウングレード

このセクションは、ポリシーを使用したスマートライセンシングがデフォルトですでに有効になっているソフトウェアバージョンで新しく購入した製品インスタンスがあり、ポリシーを使用したスマートライセンシングがサポートされていないソフトウェアバージョンにダウングレードする場合に該当します。

ダウングレードの結果は、ポリシーを使用したスマートライセンシング環境での操作中に信頼コードがインストールされたかどうかによって異なります。ダウングレード先のリリースによっては、さらにアクションが必要になる場合があります。

ポリシーを使用したスマートライセンシング環境で実装したトポロジが「CSSMに直接接続」である場合、トポロジ実装の一部として信頼コードが必要であるため、信頼コードのインストールが想定または仮定されます。他のトポロジでは、信頼の確立は必須ではありません。そのため、他のトポロジのいずれかを使用する製品インスタンスをダウングレードすると、スマートライセンシング環境で適用される手順に従って、ライセンスを登録済みおよび承認済みの状態に復元する必要が生じます。以下の表 12 : スマートライセンシングへの新規展開のダウングレードの結果とアクション （162 ページ）を参照してください。

表 *12 :* スマートライセンシングへの新規展開のダウングレードの結果とアクション

ポリシーを使用したスマートライセンシング環境で	以下にダウングレードした場合…	結果と追加のアクション
CSSM に直接接続され、信頼が確立されたスタンドアロン製品インスタンス。	Cisco IOS XE Amsterdam 17.3.1 または Cisco IOS XE Gibraltar 16.12.x の Cisco IOS XE Gibraltar 16.12.4 以降のリリース または Cisco IOS XE Fuji 16.9.x の Cisco IOS XE Fuji 16.9.6 以降のリリース	これ以上の操作は不要です。 製品インスタンスは、ダウングレード後に CSSM からの信頼を更新しようとします。 更新が正常に完了すると、ライセンスは登録済みの状態になり、以前のバージョンのスマートライセンシングが製品インスタンスで有効になります。
	スマートライセンシングをサポートするその他のリリース（上の行に記載されているものを除く）	アクションが必要です。製品インスタンスを再登録する必要があります。 CSSM Web UI で ID トークンを生成し、製品インスタンスで、グローバル コンフィギュレーション モードで **license smart register idtoken** *idtoken* コマンドを設定します。
CSSM に直接接続され、信頼が確立された高可用性セットアップ。	スマートライセンシングをサポートするすべてのリリース	アクションが必要です。製品インスタンスを再登録する必要があります。 CSSM Web UI で ID トークンを生成し、製品インスタンスで、グローバル コンフィギュレーション モードで **license smart register idtoken** *idtoken* **all** コマンドを設定します。
その他のトポロジ。（CSLU を介した CSSM への接続、CSLU は CSSM から切断、CSSM への接続なし、CSLU なし）	スマートライセンシングをサポートするすべてのリリース	アクションが必要です。 スマートライセンシング環境で適用される手順に従って、ライセンスを登録済みおよび承認済みの状態に復元します。

アップグレード後のダウングレード

ポリシーを使用したスマートライセンシングをサポートするソフトウェアバージョンにアップグレードした後、以前のライセンシングモデルのいずれかにダウングレードしても、ライセンスの使用は変更されず、製品インスタンスで設定した製品機能は維持されます。ポリシーを使用したスマートライセンシングで使用可能な機能のみが使用できなくなります。以前のライセンシングモデルへの復帰の詳細については、以下の対応するセクションを参照してください。

ポリシーを使用したスマートライセンシングへのアップグレード後のスマートライセンシングへのダウングレード

ダウングレードの結果は、ポリシーを使用したスマートライセンシング環境での操作中に信頼コードがインストールされたかどうかによって異なります。ダウングレード先のリリースによっては、さらにアクションが必要になる場合があります。「表 13 : ポリシーを使用したスマートライセンシングへのアップグレード後のスマートライセンシングへのダウングレードの結果とアクション （163 ページ）」を参照してください。

表 *13:*ポリシーを使用したスマートライセンシングへのアップグレード後のスマートライセンシングへのダウングレードの結果とアクション

ポリシーを使用したスマートライセンシング環境で	以下にダウングレードした場合...	結果と追加のアクション
CSSM に直接接続され、信頼が確立されたスタンドアロン製品インスタンス。	Cisco IOS XE Amsterdam 17.3.1 または Cisco IOS XE Gibraltar 16.12.x の Cisco IOS XE Gibraltar 16.12.4 以降のリリース または Cisco IOS XE Fuji 16.9.x の Cisco IOS XE Fuji 16.9.6 以降のリリース	これ以上の操作は不要です。 システムは信頼コードを認識し、元の登録済みIDトークンに変換します。これにより、ライセンスはAUTHORIZEDおよびREGISTEREDの状態に戻ります。
	スマートライセンシングをサポートするその他のリリース（上の行に記載されているものを除く）	アクションが必要です。製品インスタンスを再登録する必要があります。 CSSM Web UI で ID トークンを生成し、製品インスタンスで、グローバル コンフィギュレーション モードで **license smart register idtoken***idtoken* コマンドを設定します。

ポリシーを使用したスマートライセンシング環境で	以下にダウングレードした場合…	結果と追加のアクション
CSSM に直接接続され、信頼が確立された高可用性セットアップ。	スマートライセンシングをサポートするすべてのリリース	アクションが必要です。製品インスタンスを再登録する必要があります。 CSSM Web UI で ID トークンを生成し、製品インスタンスで、グローバル コンフィギュレーション モードで **license smart register idtoken** *idtoken* **all** コマンドを設定します。
その他のトポロジ（CSLU を介したCSSMへの接続、CSLU は CSSM から切断、CSSM への接続なし、CSLU なし）	スマートライセンシングをサポートするすべてのリリース	アクションが必要です。 スマートライセンシング環境で適用される手順に従って、ライセンスを登録済みおよび承認済みの状態に復元します。

（注） スマートライセンシング環境で評価状態または期限切れ状態になっていたライセンスは、ダウングレード後に同じ状態に戻ります。

ポリシーを使用したスマートライセンシングへのアップグレード後の SLR へのダウングレード

SLRに戻すのに必要な操作は、イメージのダウングレードのみです。ライセンスは予約済みおよび承認済みのままになります。これ以上の操作は必要ありません。

ただし、ポリシーを使用したスマートライセンシング環境で SLR に戻した場合は、サポートされているリリースで、必要に応じて SLR を取得するプロセスを繰り返す必要があります。

RTU へのダウングレード

RTU に戻すのに必要な操作は、イメージのダウングレードのみです。

RTUライセンシング環境で評価状態または期限切れ状態であったライセンスは、ダウングレード後に同じ状態に戻ります。

ポリシーを使用したスマートライセンシングの設定方法：トポロジ別のワークフロー

このセクションでは、トポロジを実装する最も簡単で迅速な方法について説明します。

（注） これらのワークフローは、新規展開のみに該当します。既存のライセンシングモデルから移行する場合は、ポリシーを使用したスマートライセンシングへの移行（182ページ）を参照してください。

トポロジのワークフロー：CSLU を介して CSSM に接続

製品インスタンス開始型通信とCSLU開始型通信のどちらを実装するかに応じて、対応する一連のタスクを実行します。

- 製品インスタンス開始型通信の場合のタスク
- CSLU 開始型通信の場合のタスク

製品インスタンス開始型通信の場合のタスク

CSLU のインストール→CSLU の環境設定→製品インスタンスの設定

1. ***CSLU* のインストール**

   タスクの実行場所：Windows ホスト（ラップトップ、デスクトップ、または仮想マシン（VM））

   [Smart Software Manager]https://software.cisco.com/download/home/286285506/type> [Smart Licensing Utility] からファイルをダウンロードします。

   インストールとセットアップの詳細については、『Cisco Smart License Utility Quick Start Setup Guide』を参照してください。

2. ***CSLU* の環境設定**

   タスクの実行場所：CSLU インターフェイス

   1. シスコへのログイン（CSLU インターフェイス）（210 ページ）
   2. スマートアカウントとバーチャルアカウントの設定（CSLUインターフェイス）（211ページ）
   3. CSLU での製品開始型製品インスタンスの追加（CSLU インターフェイス）（211 ページ）

3. **製品インスタンスの設定**

   タスクが実行される場所：製品インスタンス

   1. 製品インスタンス開始型通信のネットワーク到達可能性の確認（212 ページ）
   2. 転送タイプが **cslu** に設定されていることを確認します。

CSLU がデフォルトの転送タイプです。別のオプションを設定した場合は、グローバル コンフィギュレーション モードで **license smart transport cslu** コマンドを入力します。構成ファイルへの変更を保存します。

```
Device(config)# license smart transport cslu
Device(config)# exit
Device# copy running-config startup-config
```

3. CSLU の検出方法を指定します（1 つ選択）

   • オプション 1：

   No action required.`cslu-local` のゼロタッチ DNS ディスカバリ用に設定されたネームサーバー

   ここでは、DNS を設定してあり（ネームサーバーの IP アドレスが製品インスタンスで設定されている）、ホスト名 `cslu-local` が CSLU IP アドレスにマッピングされているエントリが DNS サーバーにある場合、それ以上のアクションは不要です。製品インスタンスは、ホスト名 `cslu-local` を自動的に検出します。

   • オプション 2：

   No action required.`cslu-local.<domain>` のゼロタッチ DNS ディスカバリ用に設定されたネームサーバーとドメイン

   ここでは、DNS を設定してあり（ネームサーバーの IP アドレスとドメインが製品インスタンスで設定されている）、`cslu-local.<domain>` が CSLU IP アドレスにマッピングされているエントリが DNS サーバーにある場合、それ以上のアクションは不要です。製品インスタンスは、ホスト名 cslu-local を自動的に検出します。

   • オプション 3：

   CSLU に特定の URL を設定します。

   グローバル コンフィギュレーション モードで **license smart url cslu** *http://<cslu_ip_or_host>:8182/cslu/v1/pi* コマンドを入力します。`<cslu_ip_or_host>` には、CSLU をインストールした Windows ホストのホスト名や IP アドレスを入力します。8182 はポート番号であり、CSLU が使用する唯一のポート番号です。

   ```
   Device(config)# license smart url cslu http://192.168.0.1:8182/cslu/v1/pi
   Device(config)# exit
   Device# copy running-config startup-config
   ```

4. **承認コードのインストール（該当する場合のみ）**

   タスクが実行される場所：製品インスタンス

   輸出規制ライセンスは、Cisco Catalyst アクセス、コア、およびアグリゲーションスイッチの特定のモデルでのみサポートされます（承認コード（143 ページ）を参照）。輸出規制ライセンスを使用する場合は、サポートされているプラットフォームで次のタスク：SLAC の手動要求と自動インストール （244 ページ）を実行します。

結果：

製品インスタンスは通信を開始すると、ポリシーに従って、スケジュールされた時刻に最初の RUM レポートを自動的に送信します。これをトリガーする **license smart sync** 特権EXECコマンドを入力することもできます。この最初のレポートとともに、必要に応じて、信頼コード要求を送信します。CSLU は RUM レポートを CSSM に転送し、信頼コードを含む ACK を取得します。ACK は、製品インスタンスが次回 CSLU に接続したときに製品インスタンスに適用されます。

製品インスタンスがこの情報をいつ送信するかを確認するには、特権 EXEC モードで **show license all** コマンドを入力し、出力で `Next report push:` フィールドの日付を確認します。

ブートレベルライセンスを変更する場合は、基本ライセンスまたはアドオンライセンスの設定（266 ページ）を参照してください。

承認コードを返す場合は、承認コードの返却 （251 ページ）を参照してください。

CSLU 開始型通信の場合のタスク

CSLU のインストール → CSLU の設定 → 製品インスタンスの設定 → 承認コードのインストール（該当する場合のみ） → 使用状況の同期

1. ***CSLU* のインストール**

   タスクの実行場所：Windows ホスト（ラップトップ、デスクトップ、または仮想マシン（VM））

   [Smart Software Manager]https://software.cisco.com/download/home/286285506/type> [Smart Licensing Utility] からファイルをダウンロードします。

   インストールとセットアップの詳細については、『Cisco Smart License Utility Quick Start Setup Guide』を参照してください。

2. ***CSLU* の環境設定**

   タスクの実行場所：CSLU インターフェイス

   1. シスコへのログイン（CSLU インターフェイス） （210 ページ）
   2. スマートアカウントとバーチャルアカウントの設定（CSLUインターフェイス） （211 ページ）
   3. CSLU での CSLU 開始型製品インスタンスの追加（CSLU インターフェイス） （213 ページ）

3. **製品インスタンスの設定**

   タスクが実行される場所：製品インスタンス

   CSLU 開始型通信のネットワーク到達可能性の確認 （216 ページ）

4. **承認コードのインストール（該当する場合のみ）**

   タスクの実行場所：CSLU インターフェイスと CSSM Web UI

輸出規制ライセンスは、Cisco Catalyst アクセス、コア、およびアグリゲーションスイッチの特定のモデルでのみサポートされます（承認コード （143 ページ）を参照）。輸出規制ライセンスを使用する場合は、サポートされているプラットフォームで次のタスクを実行します。

1. SLAC の手動要求と自動インストール （244 ページ）
2. 1 つ以上の製品インスタンスの SLAC の要求（CSLU インターフェイス） （221 ページ）
3. CSSM からの SLAC の生成とファイルへのダウンロード （249 ページ）
4. CSSM からのインポート（CSLU インターフェイス） （216 ページ）

5. **使用状況の同期**

   タスクの実行場所：CSLU インターフェイス

   使用状況レポートの収集：CSLU 開始（CSLU インターフェイス） （214 ページ）

結果：

CSLU が現在シスコにログインしているため、レポートは CSSM の関連するスマートアカウントとバーチャルアカウントに自動的に送信され、CSSM は CSLU と製品インスタンスに確認応答を送信します。この最初のレポートとともに、CSLU は承認コード要求を CSSM に送信します（該当する場合）。CSSM から ACK を取得し、インストールのために製品インスタンスに送り返します。

ブートレベルライセンスを変更する場合は、基本ライセンスまたはアドオンライセンスの設定（266 ページ）を参照してください。

承認コードを返す場合は、承認コードの返却 （251 ページ）を参照してください。

トポロジのワークフロー：CSSM に直接接続

スマートアカウントのセットアップ→製品インスタンスの設定→CSSM による信頼の確立→承認コードのインストール（該当する場合のみ）

1. **スマートアカウントのセットアップ**

   タスクが実行される場所：CSSM Web UI、https://software.cisco.com/

   スマートアカウントと必要なバーチャルアカウントへの適切なアクセス権を持つユーザーロールがあることを確認します。

2. **製品インスタンスの設定**

   タスクが実行される場所：製品インスタンス

   1. CSSM への製品インスタンス接続の設定：CSSM への接続の設定 （222 ページ）
   2. 接続方法と転送タイプの設定（1 つ選択）

      • オプション 1：

スマート転送：転送タイプを **smart** に設定し、対応する URL を設定します。

転送モードが **license smart transport smart** に設定されている場合は、**license smart url default** を設定すると、スマート URL（https://smartreceiver.cisco.com/licservice/license）が自動的に設定されます。構成ファイルへの変更を保存します。

```
Device(config)# license smart transport smart
Device(config)# license smart url default
Device(config)# exit
Device# copy running-config startup-config
```

- オプション 2：

  HTTPS プロキシを介してスマートトランスポートを設定します。HTTPS プロキシを介したスマート転送の設定 （224 ページ） を参照してください

- オプション 3：

  ダイレクトクラウドアクセス用に Call Home サービスを設定します。「ダイレクトクラウドアクセス用の Call Home サービスの設定 （226 ページ）」を参照してください。

- オプション 4：

  HTTPS プロキシを介したダイレクトクラウドアクセス用に Call Home サービスを設定します。「HTTPS プロキシサーバーを介したダイレクトクラウドアクセス用の Call Home サービスの設定 （229 ページ）」を参照してください。

3. *CSSM* との信頼の確立

タスクが実行される場所：CSSM Web UI、次に製品インスタンス

1. 所有するバーチャルアカウントごとに 1 つのトークンを生成します。1 つのバーチャルアカウントに属するすべての製品インスタンスに同じトークンを使用できます（CSSM からの信頼コード用新規トークンの生成 （257 ページ））。
2. トークンをダウンロードしたら、製品インスタンスに信頼コードをインストールできます（信頼コードのインストール （258 ページ））。

4. 承認コードのインストール（該当する場合のみ）

タスクが実行される場所：製品インスタンス

輸出規制ライセンスは、Cisco Catalyst アクセス、コア、およびアグリゲーションスイッチの特定のモデルでのみサポートされます（承認コード （143 ページ）を参照）。輸出規制ライセンスを使用する場合は、サポートされているプラットフォームで次のタスク：SLAC の手動要求と自動インストール （244 ページ）を実行します。

結果：

信頼を確立した後、CSSMはポリシーを返します。ポリシーは、そのバーチャルアカウントのすべての製品インスタンスに自動的にインストールされます。ポリシーは、製品インスタンスが使用状況をレポートするかどうか、およびその頻度を指定します。

より頻繁にレポートを作成するようにレポート間隔を変更する場合は、製品インスタンスで、グローバル コンフィギュレーション モードで **license smart usage interval** コマンドを設定します。シンタックスの詳細については、対応するリリースのコマンドリファレンスで *license smart (privileged EXEC)* コマンドを参照してください。

ブートレベルライセンスを変更する場合は、基本ライセンスまたはアドオンライセンスの設定（266 ページ）を参照してください。

承認コードを返す場合は、承認コードの返却 （251 ページ）を参照してください。

トポロジのワークフロー：コントローラを介して CSSM に接続

コントローラとして Cisco DNA Center を展開するには、次のワークフローを実行します。

製品インスタンスの設定 → Cisco DNA Center の設定

1. **製品インスタンスの設定**

   タスクの実行場所：製品インスタンス

   NETCONF を有効にします。Cisco DNA Center は NETCONF プロトコルを使用して設定をプロビジョニングし、製品インスタンスから必要な情報を取得します。したがって、これを容易にするために製品インスタンスで NETCONF を有効にする必要があります。

   詳細については、『Programmability Configuration Guide, Cisco IOS XE Amsterdam 17.3.x』を参照してください。このガイドの「Model-Driven Programmability」の「NETCONF Protocol」を確認します。

2. **Cisco DNA Center の設定**

   タスクの実行場所：Cisco DNA Center GUI

   次に、実行する必要のあるタスクの概要と、付属のドキュメントリファレンスを示します。このドキュメントには、Cisco DNA Center GUI で実行する必要がある詳細な手順が示されています。

   1. スマートアカウントとバーチャルアカウントを設定します。

      CSSM Web UI へのログインに使用するのと同じログインクレデンシャルを入力します。これにより、Cisco DNA Center は CSSM との接続を確立できます。

      必要なリリース（リリース 2.2.2 以降）の『Cisco DNA Center Administrator Guide』の「Manage Licenses」の「Set Up License Manager」を参照してください。

   2. 必要な製品インスタンスを Cisco DNA Center インベントリに追加してサイトに割り当てます。

      これにより、Cisco DNA Center は、要求されている証明書を含む必要な設定をプッシュして、Smart Licensing Using Policy が予想どおりに機能するようにします。

必要なリリース（リリース 2.2.2 以降）の『Cisco DNA Center User Guide』の「Display Your Network Topology」の「Assign Devices to a Site」を参照してください。

結果：

トポロジを実装したら、Cisco DNA Center で最初のアドホックレポートをトリガーし、スマートアカウントとバーチャルアカウント、および製品インスタンス間のマッピングを確立する必要があります。必要なリリース（リリース 2.2.2 以降）の『Cisco DNA Center Administrator Guide』で「Manage Licenses」の「Upload Resource Utilization Details to CSSM」を参照してください。これが完了すると、Cisco DNA Center はレポートポリシーに基づいて後続のレポートを処理します。

複数のポリシーが使用可能な場合、Cisco DNA Centerは 最も短いレポート間隔を維持します。この間隔はより頻繁に（より短い間隔で）報告するようにのみ変更できます。必要なリリース（リリース 2.2.2 以降）の『Cisco DNA Center Administrator Guide』の「Manage Licenses」の「Modify License Policy」を参照してください。

この後にライセンスレベルを変更する場合は、必要なリリース（リリース 2.2.2 以降）の『Cisco DNA Center Administrator Guide』の「Manage Licenses」の「Change License Level」を参照してください。

トポロジのワークフロー：CSLU は CSSM から切断

製品インスタンス開始型通信またはCSLU開始型通信のどちらの方法を実装するかによって異なります。以下の対応するタスク一覧を実行します。

- 製品インスタンス開始型通信の場合のタスク
- CSLU 開始型通信の場合のタスク

製品インスタンス開始型通信の場合のタスク

CSLU のインストール → CSLU の設定 → 製品インスタンスの設定 → 承認コードのインストール（該当する場合のみ） → 使用状況の同期

1. ***CSLU* のインストール**

   タスクの実行場所：Windows ホスト（ラップトップ、デスクトップ、または仮想マシン（VM））

   [Smart Software Manager]https://software.cisco.com/download/home/286285506/type> [Smart Licensing Utility] からファイルをダウンロードします。

   インストールとセットアップの詳細については、『Cisco Smart License Utility Quick Start Setup Guide』を参照してください。

2. ***CSLU* の環境設定**

   タスクの実行場所：CSLU インターフェイス

1. CSLU の [Preferences] タブで、[Cisco Connectivity] トグルスイッチを**オフ**にします。フィールドが「Cisco Is Not Available」に切り替わります。
2. スマートアカウントとバーチャルアカウントの設定（CSLUインターフェイス）（211ページ）
3. CSLUでの製品開始型製品インスタンスの追加（CSLUインターフェイス）（211ページ）

3. 製品インスタンスの設定

タスクが実行される場所：製品インスタンス

1. 製品インスタンス開始型通信のネットワーク到達可能性の確認（212 ページ）
2. 転送タイプが **cslu** に設定されていることを確認します。

   CSLU がデフォルトの転送タイプです。別のオプションを設定した場合は、グローバル コンフィギュレーション モードで **license smart transport cslu** コマンドを入力します。構成ファイルへの変更を保存します。

   ```
   Device(config)# license smart transport cslu
   Device(config)# exit
   Device# copy running-config startup-config
   ```

3. CSLU の検出方法を指定します（1 つ選択）

   - オプション 1：

     No action required.`cslu-local` のゼロタッチ DNS ディスカバリ用に設定されたネームサーバー

     ここでは、DNS を設定してあり（ネームサーバーの IP アドレスが製品インスタンスで設定されている）、ホスト名 `cslu-local` が CSLU IP アドレスにマッピングされているエントリが DNS サーバーにある場合、それ以上のアクションは不要です。製品インスタンスは、ホスト名 `cslu-local` を自動的に検出します。

   - オプション 2：

     No action required.`cslu-local.<domain>` のゼロタッチ DNS ディスカバリ用に設定されたネームサーバーとドメイン

     ここでは、DNS を設定してあり（ネームサーバーの IP アドレスとドメインが製品インスタンスで設定されている）、`cslu-local.<domain>` が CSLU IP アドレスにマッピングされているエントリが DNS サーバーにある場合、それ以上のアクションは不要です。製品インスタンスは、ホスト名 cslu-local を自動的に検出します。

   - オプション 3：

     CSLU に特定の URL を設定します。

     グローバル コンフィギュレーション モードで **license smart url cslu** *http://<cslu_ip_or_host>:8182/cslu/v1/pi* コマンドを入力します。`<cslu_ip_or_host>` には、CSLU をインストールした Windows ホストのホスト名や IP アドレスを入力します。8182 はポート番号であり、CSLU が使用する唯一のポート番号です。

```
Device(config)# license smart url cslu http://192.168.0.1:8182/cslu/v1/pi
Device(config)# exit
Device# copy running-config startup-config
```

4. 承認コードのインストール（該当する場合のみ）

   タスクの実行場所：製品インスタンスと CSSM Web UI

   輸出規制ライセンスは、Cisco Catalyst アクセス、コア、およびアグリゲーションスイッチの特定のモデルでのみサポートされます（承認コード（143 ページ）を参照）。輸出規制ライセンスを使用する場合は、サポートされているプラットフォームで次のタスクを実行します。

   1. SLAC の手動要求と自動インストール （244 ページ）
   2. 1 つ以上の製品インスタンスの SLAC の要求（CSLU インターフェイス） （221 ページ）
   3. CSSM からの SLAC の生成とファイルへのダウンロード （249 ページ）
   4. CSSM からのインポート（CSLU インターフェイス） （216 ページ）

5. 使用状況の同期

   タスクの実行場所：CSLU と CSSM

   製品インスタンスは通信を開始すると、ポリシーに従って、スケジュールされた時刻に最初の RUM レポートを自動的に送信します。これをトリガーする **license smart sync** 特権 EXEC コマンドを入力することもできます。この最初のレポートとともに、必要に応じて、UDI に関連付けられた信頼コード要求を送信します。CSLU は CSSM から切断されているため、次のタスクを実行して RUM レポートを CSSM に送信します。

   1. CSSM へのエクスポート（CSLU インターフェイス） （215 ページ）
   2. CSSM への使用状況データのアップロードと ACK のダウンロード （260 ページ）
   3. CSSM からのインポート（CSLU インターフェイス） （216 ページ）

結果：

CSSM からインポートした ACK に信頼コードが含まれます（要求した場合）。ACK は、製品インスタンスが次回 CSLU に接続したときに製品インスタンスに適用されます。

製品インスタンスが次にいつ RUM レポートを送信するかを確認するには、特権 EXEC モードで **show license all** コマンドを入力し、出力の [Next report push] フィールドの日付を確認します。

ブートレベルライセンスを変更する場合は、基本ライセンスまたはアドオンライセンスの設定（266 ページ）を参照してください。

承認コードを返す場合は、承認コードの返却 （251 ページ）を参照してください。

CSLU 開始型通信の場合のタスク

CSLU のインストール → CSLU の設定 → 製品インスタンスの設定 → 承認コードのインストール（該当する場合のみ） → 使用状況の同期

1. ***CSLU* のインストール**

   タスクの実行場所：Windows ホスト（ラップトップ、デスクトップ、または仮想マシン（VM））

   [Smart Software Manager]https://software.cisco.com/download/home/286285506/type> [Smart Licensing Utility] からファイルをダウンロードします。

   インストールとセットアップの詳細については、『Cisco Smart License Utility Quick Start Setup Guide』を参照してください。

2. ***CSLU* の環境設定**

   タスクの実行場所：CSLU

   1. CSLU の [Preferences] タブで、[Cisco Connectivity] トグルスイッチを**オフ**にします。フィールドが「Cisco Is Not Available」に切り替わります。
   2. スマートアカウントとバーチャルアカウントの設定（CSLUインターフェイス）（211 ページ）
   3. CSLU での CSLU 開始型製品インスタンスの追加（CSLU インターフェイス）（213 ページ）
   4. 使用状況レポートの収集：CSLU 開始（CSLU インターフェイス）（214 ページ）

3. **製品インスタンスの設定**

   タスクの実行場所：製品インスタンス

   CSLU 開始型通信のネットワーク到達可能性の確認（216 ページ）

4. **承認コードのインストール（該当する場合のみ）**

   タスクが実行される場所：製品インスタンス

   輸出規制ライセンスは、Cisco Catalyst アクセス、コア、およびアグリゲーションスイッチの特定のモデルでのみサポートされます（承認コード（143 ページ）を参照）。輸出規制ライセンスを使用する場合は、サポートされているプラットフォームで次のタスクを実行します。

   1. SLAC の手動要求と自動インストール（244 ページ）
   2. 1 つ以上の製品インスタンスの SLAC の要求（CSLU インターフェイス）（221 ページ）
   3. CSSM からの SLAC の生成とファイルへのダウンロード（249 ページ）
   4. CSSM からのインポート（CSLU インターフェイス）（216 ページ）

5. **使用状況の同期**

   タスクの実行場所：CSLU と CSSM

   製品インスタンスから使用状況データを収集します。CSLU は CSSM から切断されるため、後で CSLU が製品インスタンスから収集した使用状況データをファイルに保存します。次に、シスコに接続されているワークステーションからファイルを CSSM にアップロードします。この後、CSSM から ACK をダウンロードします。CSLU がインストールされて製品インスタンスに接続されているワークステーションで、ファイルを CSLU にアップロードします。

   1. CSSM へのエクスポート（CSLU インターフェイス）（215 ページ）
   2. CSSM への使用状況データのアップロードと ACK のダウンロード（260 ページ）
   3. CSSM からのインポート（CSLU インターフェイス）（216 ページ）

結果：

CSLU が次に更新を実行するときに、アップロードされた ACK が製品インスタンスに適用されます。

ブートレベルライセンスを変更する場合は、基本ライセンスまたはアドオンライセンスの設定（266 ページ）を参照してください。

承認コードを返す場合は、承認コードの返却（251 ページ）を参照してください。

トポロジのワークフロー：CSSM への接続なし、CSLU なし

他のコンポーネントへの接続を設定する必要がないため、トポロジの設定に必要なタスクのリストは短くなります。このトポロジを実装した後に必要な使用状況レポートを作成する方法については、ワークフローの最後にある「結果」セクションを参照してください。

製品インスタンスの設定→承認コードのインストール（該当する場合のみ）

1. **製品インスタンスの設定**

   タスクの実行場所：製品インスタンス

   転送タイプを**オフ**に設定します。

   グローバル コンフィギュレーション モードで **license smart transport off** コマンドを入力します。構成ファイルへの変更を保存します。

   ```
   Device(config)# license smart transport off
   Device(config)# exit
   Device# copy running-config startup-config
   ```

2. **承認コードのインストール（該当する場合のみ）**

   タスクが実行される場所：CSSM Web UI および製品インスタンス

   輸出規制ライセンスは、Cisco Catalyst アクセス、コア、およびアグリゲーションスイッチの特定のモデルでのみサポートされます（承認コード（143 ページ）を参照）。輸出規制

ライセンスを使用する場合は、サポートされているプラットフォームで次のタスクを実行します。

1. CSSM からの SLAC の生成とファイルへのダウンロード （249 ページ）。
2. 製品インスタンスへのファイルのインストール （261 ページ）。

結果：

製品インスタンスからのすべての通信を無効にします。ライセンスの使用状況をレポートするには、RUM レポートを（製品インスタンスの）ファイルに保存してから、CSSM にアップロードする必要があります（インターネットとシスコに接続されているワークステーションからアップロード）。

1. RUM レポートの生成と保存

   license smart save usage コマンドは特権 EXEC モードで入力します。次の例では、すべての RUM レポートがファイル `all_rum.txt` で製品インスタンスのフラッシュメモリに保存されます。この例では、ファイルはまずブートフラッシュに保存され、次に TFTP の場所にコピーされます。

   ```
   Device# license smart save usage all bootflash:all_rum.txt
   Device# copy bootflash:all_rum.txt tftp://10.8.0.6/all_rum.txt
   ```

2. 使用状況データを CSSM にアップロード： CSSM への使用状況データのアップロードと ACK のダウンロード （260 ページ）
3. ACK を製品インスタンスにインストール： 製品インスタンスへのファイルのインストール （261 ページ）

ブートレベルライセンスを変更する場合は、基本ライセンスまたはアドオンライセンスの設定（266 ページ）を参照してください。

承認コードを返す場合は、承認コードの返却 （251 ページ）を参照してください。

トポロジのワークフロー：SSM オンプレミス展開

製品インスタンス開始型通信（プッシュ）を実装するか、または SSM オンプレミス開始型通信（プル）を実装するかによって、対応するタスクの手順を実行します。

製品インスタンス開始型通信の場合のタスク

SSM オンプレミスのインストール → 製品インスタンスの追加と検証（該当する場合のみ） → 製品インスタンスの設定 → 使用状況の最初の同期

1. ***SSM* オンプレミスのインストール**

   タスクの実行場所：Cisco UCS C220 M3 ラックサーバーなどの物理サーバー、または必要な要件を満たしているハードウェアベースのサーバー。

   Smart Software Manager の [Smart Software Manager On-Prem] からファイルをダウンロードします。

インストールのヘルプについては、『*Cisco Smart Software On-Prem Installation Guide*』と『*Cisco Smart Software On-Prem User Guide*』を参照してください。

SSM オンプレミスを展開し、SSM オンプレミスで共通名を設定し（[Security Widgets] > [Certificates]）、NTP サーバーを同期し（[Settings] ウィジェット > [Time Settings]）、SSM オンプレミスアカウントを作成して登録し、CSSM のスマートアカウントとバーチャルアカウントと同期（[Synchronization] ウィジェット）したら、インストールが完了します。

（注） [On-Prem Licensing Workspace] のライセンス機能は、ローカルアカウントを作成し、登録し、CSSM のスマートアカウントと同期するまではグレー表示になります。CSSM とのローカルアカウントの同期は、SSM オンプレミスインスタンスを CSSM に認識させるためであり、次に示す「**4. 使用状況の最初の同期」で実行する使用状況の同期とは異なります。**

2. 製品インスタンスの追加と検証

タスクの実行場所：SSM オンプレミス UI

この手順により、製品インスタンスが検証され、CSSM の該当するスマートアカウントとバーチャルアカウントにマッピングされます。この手順は、次の場合にのみ必要です。

- 製品インスタンスを CSSM で報告する前に、SSM オンプレミスで追加および検証する場合（セキュリティを強化するため）。
- 使用前に承認が必要なライセンスを使用する場合（適用タイプ：適用（エンフォースメント）または輸出規制）：次の手順 3 d で必要な SLAC を要求する前に、このような製品インスタンスを SSM オンプレミスに追加する必要があります。
- （デフォルトのローカル バーチャル アカウントに加えて）ローカル バーチャル アカウントを SSM オンプレミスで作成した場合。この場合は、SSM オンプレミスが CSSM の正しいライセンスプールに使用状況を報告できるように、SSM オンプレミスにこれらのローカル バーチャル アカウントの製品インスタンスのスマートアカウント情報とバーチャルアカウント情報を提供する必要があります。

1. スマートアカウントとバーチャルアカウントの割り当て（SSM オンプレミス UI）（230 ページ）
2. デバイスの検証（SSM オンプレミス UI）（231 ページ）

（注） 製品インスタンスが NAT 設定にある場合は、デバイス検証を有効にするときに NAT 設定のサポートも有効にします。両方のトグルスイッチが同じウィンドウにあります。

3. 製品インスタンスの設定

タスクの実行場所：製品インスタンスと SSM オンプレミス UI

特権 EXEC モードで **copy running-config startup-config** コマンドを入力して、製品インスタンスの設定変更を必ず保存してください。

1. 製品インスタンス開始型通信のネットワーク到達可能性の確認 （232 ページ）
2. トランスポート URL の取得（SSM オンプレミス UI） （234 ページ）
3. 転送タイプ、URL、およびレポート間隔の設定 （262 ページ）

   CSLU と SSM オンプレミスのトランスポートタイプ設定は同じですが（グローバル コンフィギュレーション モードの **license smart transport cslu** コマンド）、URL が異なります。
4. 輸出規制ライセンスは、Cisco Catalyst アクセス、コア、およびアグリゲーションスイッチの特定のモデルでのみサポートされます（承認コード （143 ページ）を参照）。サポートされているプラットフォームで輸出規制対象ライセンスを使用する場合にのみ、次のサブステップ：承認コード要求の送信（SSM オンプレミス UI） （243 ページ）および SLAC の手動要求と自動インストール （244 ページ）を実行します。

4. 使用状況の最初の同期

タスクの実行場所：製品インスタンス、SSM オンプレミス UI、CSSM

1. 製品インスタンスを SSM オンプレミスと同期します。

   製品インスタンスに **license smart sync** {**all** | **local**} コマンドを特権 EXEC モードで入力します。これにより、SSM オンプレミスと製品インスタンスが同期され、保留中のデータが送受信されます。

   ```
   Device(config)# license smart sync local
   ```

   これは、SSM オンプレミス UI で確認できます。ログインして、[Smart Licensing] ワークスペースを選択します。[Inventory] > [SL Using Policy] タブに移動します。対応する製品インスタンスの [Alerts] 列に、「Usage report from product instance」というメッセージが表示されます。

（注） 上記の手順 2（製品インスタンスの追加と検証）を実行していない場合、このサブ手順を実行すると、製品インスタンスが SSM オンプレミスのデータベースに追加されます。

2. 使用状況情報を CSSM と同期します（いずれかを選択）。
   - オプション 1：

     SSM オンプレミスが CSSM に接続されている場合：SSM オンプレミス UI の [Smart Licensing] ワークスペースで、[Reports] > [Usage Schedules] > [Synchronize now with Cisco] に移動します。
   - オプション 2：

     SSM オンプレミスが CSSM に接続されていません。使用状況データのエクスポートとインポート（SSM オンプレミス UI） （235 ページ）を参照してください。

結果：

使用状況の最初の同期が完了しました。製品インスタンスとライセンス使用状況情報が SSM オンプレミスに表示されるようになりました。

後続のレポートには、次のオプションが含まれています。

- 製品インスタンスと SSM オンプレミスとの間でデータを同期するには、次の手順を実行します。
  - レポート間隔を設定して、製品インスタンスと SSM オンプレミスとの間の定期的な同期をスケジュールします。グローバル コンフィギュレーション モードで **license smart usage interval** *interval_in_days* コマンドを入力します。

    製品インスタンスが次にいつRUM レポートを送信するかを確認するには、特権EXEC モードで **show license all** コマンドを入力し、出力の [Next report push:] フィールドを確認します。
  - 製品インスタンスと SSM オンプレミスとの間でアドホックまたはオンデマンドの同期を行うには、**license smart sync** 特権 EXEC コマンドを入力します。
- 使用状況情報を CSSM と同期するには、次の手順を実行します。
  - CSSM との定期的な同期をスケジュールします。SSM オンプレミス UI で、[Reports] > [Usage Schedules] > [Synchronization schedule with Cisco] に移動します。次の頻度情報を入力し、保存します。
    - [Days]：同期が実行される頻度を示します。たとえば、2 を入力すると、同期は 2 日に 1 回行われます。
    - [Time of Day]：24 時間表記法で、同期が実行される時刻を示します。たとえば、14 hours と 0 minutes を入力すると、ローカルタイムゾーンの午後 2 時（1400）に同期が行われます。
  - レポートに必要なファイルのアップロードとダウンロードを実行します（使用状況データのエクスポートとインポート（SSM オンプレミス UI）（235 ページ））。

ブートレベルライセンスを変更する場合は、基本ライセンスまたはアドオンライセンスの設定（266 ページ）を参照してください。

承認コードを返す場合は、承認コードの返却（251 ページ）を参照してください。

SSM オンプレミスインスタンス開始型通信の場合のタスク

SSM オンプレミスのインストール → 製品インスタンスの追加 → 製品インスタンスの設定 → 使用状況の最初の同期

1. ***SSM* オンプレミスのインストール**

   タスクの実行場所：Cisco UCS C220 M3 ラックサーバーなどの物理サーバー、または必要な要件を満たしているハードウェアベースのサーバー。

Smart Software Manager の [Smart Software Manager On-Prem] からファイルをダウンロードします。

インストールのヘルプについては、『*Cisco Smart Software On-Prem Installation Guide*』と『*Cisco Smart Software On-Prem User Guide*』を参照してください。

SSM オンプレミスを展開し、SSM オンプレミスで共通名を設定し（[Security Widgets] > [Certificates]）、NTP サーバーを同期し（[Settings] ウィジェット > [Time Settings]）、SSM オンプレミスアカウントを作成して登録し、CSSM のスマートアカウントとバーチャルアカウントと同期（[Synchronization] ウィジェット）したら、インストールが完了します。

（注） [On-Prem Licensing Workspace] のライセンス機能は、ローカルアカウントを作成し、登録し、CSSM のスマートアカウントと同期するまではグレー表示になります。CSSM とのローカルアカウントの同期は、SSM オンプレミスインスタンスを CSSM に認識させるためであり、次に示す「**4. 使用状況の最初の同期」で実行する使用状況の同期とは異なります。**

2. 製品インスタンスの追加

タスクの実行場所：SSM オンプレミス UI

単一の製品インスタンスを追加するか、または複数の製品インスタンスを追加するかに応じて、対応するサブ手順（1 つ以上の製品インスタンスの追加（SSM オンプレミス UI）（236 ページ））を実行します。

3. 製品インスタンスの設定

タスクが実行される場所：製品インスタンス

特権 EXEC モードで **copy running-config startup-config** コマンドを入力して、製品インスタンスの設定変更を必ず保存してください。

1. SSM オンプレミス開始型通信のネットワーク到達可能性の確保 （237 ページ）
2. 輸出規制ライセンスは、Cisco Catalyst アクセス、コア、およびアグリゲーションスイッチの特定のモデルでのみサポートされます（承認コード （143 ページ）を参照）。サポートされているプラットフォームで輸出規制ライセンスを使用する場合にのみ、次のサブステップを実行します。承認コード要求の送信（SSM オンプレミス UI） （243 ページ）

   SSM オンプレミスが次に更新を実行するときに、アップロードされたコードが適用されます。製品インスタンスとの使用状況の最初の同期は、次の手順 4 で実行されて、その後に完了します。

4. 使用状況の最初の同期

タスクの実行場所：SSM オンプレミスと CSSM

1. 製品インスタンスから使用状況情報を取得します。

SSM オンプレミス UI で、[Reports] > [Synchronization pull schedule] > [Synchronize now with the device] に移動します。

[Alerts] 列に、「Usage report from product instance」というメッセージが表示されます。

ヒント 同期がトリガーされるまでに 60 秒かかります。進行状況を表示するには、[On-Prem Admin Workspace] に移動し、[Support Center] ウィジェットをクリックします。このウィジェットにシステムログに進行状況が表示されます。

2. 使用状況情報を CSSM と同期します（いずれかを選択）。
   - オプション 1：

     SSM オンプレミスが CSSM に接続されている場合：SSM オンプレミス UI の [Smart Licensing] ワークスペースで、[Reports] > [Usage Schedules] > [Synchronize now with Cisco] に移動します。
   - オプション 2：

     SSM オンプレミスが CSSM に接続されていません。使用状況データのエクスポートとインポート（SSM オンプレミス UI） （235 ページ）を参照してください。

結果：

使用状況の最初の同期が完了しました。製品インスタンスとライセンス使用状況情報が SSM オンプレミスに表示されるようになりました。SSM オンプレミスは ACK を製品インスタンスに自動的に返します。製品インスタンスが ACK を受信していることを確認するには、特権 EXEC モードで **show license status** コマンドを入力し、出力で [Last ACK received] フィールドの日付を確認します。

後続のレポートには、次のオプションが含まれています。

- 製品インスタンスから使用状況情報を取得するには、次の手順を実行します。
  - SSM オンプレミス UI の [Smart Licensing] ワークスペースで、[Reports] > [Usage Schedules] > [Synchronize now with Cisco）]に移動します。
  - 頻度を設定して、製品インスタンスから情報を定期的に取得するようにスケジュールします。SSM オンプレミス UI の [Smart Licensing] ワークスペースで、[Reports] > [Usage Schedules] > [Synchronisation pull schedule with the devices] に移動します。次のフィールドに値を入力します。
    - [Days]：同期が実行される頻度を示します。たとえば、2 を入力すると、同期は 2 日に 1 回行われます。
    - [Time of Day]：24 時間表記法で、同期が実行される時刻を示します。たとえば、14 hours と 0 minutes と入力すると、午後 2 時（1400）に同期が行われます。

- CSSMに接続せずに製品インスタンスから使用状況データを収集します。SSMオンプレミスUIの[Smart Licensing]ワークスペースで、[Inventory] > [SL Using Policy]タブに移動します。対応するチェックボックスを有効にして、1つ以上の製品インスタンスを選択します。[Actions for Selected...] > [Collect Usage]をクリックします。選択した製品インスタンスにオンプレミスが接続し、使用状況レポートを収集します。その後、これらの使用状況レポートはオンプレミスのローカルライブラリに保存されます。これらのレポートは、オンプレミスがシスコに接続されている場合はシスコに転送できます。また、（シスコに接続されていない場合は）[Export/Import All..] > [Export Usage to Cisco]を選択することで、使用状況の収集を手動でトリガーできます。

- 使用状況情報をCSSMと同期するには、次の手順を実行します。
  - CSSMとの定期的な同期をスケジュールします。SSMオンプレミスUIで、[Reports] > [Usage Schedules] > [Synchronization schedule with Cisco]に移動します。次の頻度情報を入力し、保存します。
    - [Days]：同期が実行される頻度を示します。たとえば、2を入力すると、同期は2日に1回行われます。
    - [Time of Day]：24時間表記法で、同期が実行される時刻を示します。たとえば、14 hoursと0 minutesと入力すると、午後2時（1400）に同期が行われます。
  - レポートに必要なファイルのアップロードとダウンロードを実行します（使用状況データのエクスポートとインポート（SSMオンプレミスUI）（235ページ））。

ブートレベルライセンスを変更する場合は、基本ライセンスまたはアドオンライセンスの設定（266ページ）を参照してください。

承認コードを返す場合は、承認コードの返却（251ページ）を参照してください。

ポリシーを使用したスマートライセンシングへの移行

ポリシーを使用したスマートライセンシングにアップグレードするには、製品インスタンスのソフトウェアバージョン（イメージ）をサポートされているバージョンにアップグレードする必要があります。

はじめる前に

ポリシーを使用したスマートライセンシングによって以前の全ライセンスモデルのさまざまな側面がどのように処理されるかを理解するため、アップグレード（158ページ）のセクションを必ずお読みください。

ポリシーを使用したスマートライセンシングは、Cisco IOS XE Amsterdam 17.3.2で導入されました。そのため、これがポリシーを使用したスマートライセンシングに最低限必要なバージョンになります。

移行前に使用していたすべてのライセンスは、アップグレード後も使用できることに注意してください。つまり、登録済みライセンスと承認済みライセンス（予約済みライセンスを含む）だけでなく、評価ライセンスもすべて移行されます。登録済みライセンスと承認済みライセンスを移行する利点は、アップグレード後も設定（トランスポートタイプの設定と、CSSMへの接続の設定、すべての証人コード）が保持されるため、移行後に実行する設定手順が少なくなります。これにより、Smart Licensing Using Policy 環境への移行がよりスムーズになります。

デバイス先行の変換は、ポリシーを使用したスマートライセンシングへの移行ではサポートされていません。

スイッチ ソフトウェアのアップグレード

アップグレードの手順については、対応するリリースノートを参照してください。一般的なリリース固有の考慮事項がある場合は、対応するリリースノートに記載されています。たとえば、Cisco IOS XE Amsterdam 17.3.2 にアップグレードするには、『*Release Notes for Cisco* <プラットフォーム名>, *Cisco IOS XE Amsterdam 17.3.x*』を参照してください。

この手順を使用して、インストールモードで、またはIn-Service Software Upgrade（ISSU）を使用してアップグレードできます（サポートされているプラットフォームおよびサポートされているリリースで実行）。

Release Notes for Cisco Catalyst 9300 シリーズ スイッチ：https://www.cisco.com/c/en/us/support/switches/catalyst-9300-series-switches/products-release-notes-list.html。「スイッチソフトウェアのアップグレード」を参照してください。ISSU は、この製品インスタンスではサポートされていません。

ソフトウェアバージョンのアップグレード後

- トポロジを実装します。

  アップグレード前の設定でトランスポートモードを使用できる場合は、アップグレード後も保持されます。評価ライセンスや、トランスポートタイプの概念が存在しないライセンスモデルの場合など、一部の場合にのみ、デフォルト（cslu）が適用されます。このような場合は、Smart Licensing Using Policy 環境で動作するように設定する前に実行する必要がある手順がいくつかある場合があります。

  アップグレード元のライセンスモデルに関係なく、アップグレード後にトポロジを変更できます。

- ライセンスの使用状況と CSSM の同期

  どのライセンスモデルからアップグレードするか、どのトポロジを実装するかに関係なく、使用状況情報をCSSMと同期します。そのためには、実装するトポロジに適用されるレポート方式に従う必要があります。この最初の同期により、使用状況の最新の情報がCSSMに反映され、カスタムポリシー（使用可能な場合）が適用されます。この同期後に適用されるポリシーは、後続のレポート要件も示します。これらのルールをアップグレードが既存ライセンスのレポートに与える影響 （159 ページ）の表にも示します。

（注） 使用状況の最初の同期が完了した後、ポリシー、またはシステムメッセージに示されている場合にのみ、レポートが必要です。

移行シナリオの例

さまざまな既存のライセンスモデルとライセンスを考慮した移行シナリオの例を示します。すべてのシナリオで、移行前と後の出力例と注意すべき CSSM Web UI の変更を（移行の成功または追加アクションのインジケータとして）示し、また、必要な移行後の手順を特定して実行する方法も示します。

（注） SSMオンプレミスでは、アップグレード関連のさまざまなアクティビティを実行する順序が重要です。したがって、このシナリオでのみ、例ではなく、移行の順序が示されています。

例：スマートライセンシングからポリシーを使用したスマートライセンシングへ

次に、スマートライセンシングからポリシーを使用したスマートライセンシングに移行する Cisco Catalyst 9500 スイッチの例を示します。これはアクティブとスタンバイを含む高可用性セットアップの例です。

- 表 14: スマートライセンシングからポリシーを使用したスマートライセンシングへ：show コマンド
- 移行後の CSSM Web UI （188 ページ）
- 移行後のレポート （191 ページ）

show コマンドは、移行の前後に確認すべき以下の重要なフィールドを抽出して出力します。

表 *14*: スマートライセンシングからポリシーを使用したスマートライセンシングへ：*show* コマンド

アップグレード前	アップグレード後
show license summary（スマートライセンシング） `Status`フィールドと `License Authorization` フィールドに、ライセンスについて `REGISTERED` および `AUTHORIZED` と表示されます。	**show license summary**（ポリシーを使用したスマートライセンシング） `Status` フィールドに、ライセンスについて、登録済みおよび承認済みではなく `IN USE` と表示されます。

アップグレード前	アップグレード後
Device# **show license summary** Smart Licensing is ENABLED Registration: **Status: REGISTERED** Smart Account: SA-Eg-Company-01 Virtual Account: SLE_Test Export-Controlled Functionality: ALLOWED Last Renewal Attempt: None Next Renewal Attempt: Mar 21 11:08:58 2021 PST License Authorization: **Status: AUTHORIZED** Last Communication Attempt: SUCCEEDED Next Communication Attempt: Oct 22 11:09:07 2020 PST License Usage: License Entitlement tag Count Status --- C9500 Network Advantage (C9500 Network Advantage) 2 AUTHORIZED C9500-DNA-16X-A (C9500-16X DNA Advantage) 2 AUTHORIZED	Device# **show license summary** License Usage: License Entitlement tag Count Status --- network-advantage (C9500 Network Advantage) 2 **IN USE** dna-advantage (C9500-16X DNA Advantage) 2 **IN USE**
show license usage（スマートライセンシング） Device# **show license usage** License Authorization: Status: AUTHORIZED on Sep 22 11:09:07 2020 PST C9500 Network Advantage (C9500 Network Advantage): Description: C9500 Network Advantage **Count: 2** Version: 1.0 **Status: AUTHORIZED** Export status: NOT RESTRICTED C9500-DNA-16X-A (C9500-16X DNA Advantage): Description: C9500-DNA-16X-A **Count: 2** Version: 1.0 **Status: AUTHORIZED** Export status: NOT RESTRICTED	**show license usage**（ポリシーを使用したスマートライセンシング） ライセンス数は変わりません。 [Enforcement Type] フィールドに NOT ENFORCED と表示されます。これは、アップグレード前に使用されていたすべてのライセンスが不適用ライセンスであったためです。 Device# **show license usage** License Authorization: Status: Not Applicable network-advantage (C9500 Network Advantage): Description: network-advantage **Count: 2** Version: 1.0 Status: IN USE Export status: NOT RESTRICTED Feature Name: network-advantage Feature Description: network-advantage **Enforcement type: NOT ENFORCED** **License type: Perpetual** dna-advantage (C9500-16X DNA Advantage): Description: C9500-16X DNA Advantage **Count: 2** Version: 1.0 Status: IN USE Export status: NOT RESTRICTED Feature Name: dna-advantage Feature Description: C9500-16X DNA Advantage **Enforcement type: NOT ENFORCED** **License type: Subscription**

show license status（スマートライセンシング）	**show license status**（ポリシーを使用したスマートライセンシング） `Transport:` フィールド：特定の転送タイプが設定されたため、アップグレード後もその設定が保持されます。 `Policy:` ヘッダーと詳細：スマートアカウントまたはバーチャルアカウントでカスタムポリシーを使用できます。これは製品インスタンスにも自動的にインストールされます。（信頼を確立した後、CSSMはポリシーを返します。その後、このポリシーが自動的にインストールされます）。 `Usage Reporting:` ヘッダー：`Next report push:` フィールドには、製品インスタンスが次の RUM レポートを CSSM に送信するタイミングについての情報が表示されます。 `Trust Code Installed:` フィールド：ID トークンが正常に変換され、信頼できる接続が CSSM で確立されたことを示します。

```
Device# show license status

Smart Licensing is ENABLED
Utility:
Status: DISABLED
Data Privacy:
Sending Hostname: yes
Callhome hostname privacy: DISABLED
Smart Licensing hostname privacy: DISABLED
Version privacy: DISABLED
Transport:
Type: Callhome
Registration:
Status: REGISTERED
Smart Account: Eg-SA-01
Virtual Account: Eg-VA-01
Export-Controlled Functionality: ALLOWED
Initial Registration: SUCCEEDED on Sep 22 11:08:58 2020 PST
Last Renewal Attempt: None
Next Renewal Attempt: Mar 21 11:08:57 2021 PST
Registration Expires: Sep 22 11:04:23 2021 PST
License Authorization:
Status: AUTHORIZED on Sep 22 11:09:07 2020 PST
Last Communication Attempt: SUCCEEDED on Sep 22 11:09:07 2020
PST
Next Communication Attempt: Oct 22 11:09:06 2020 PST
Communication Deadline: Dec 21 11:04:34 2020 PST
Export Authorization Key:
Features Authorized:
<none>
Miscellaneus:
Custom Id: <empty>
```

```
Device# show license status

Utility:
  Status: DISABLED
Smart Licensing Using Policy:
  Status: ENABLED
Data Privacy:
  Sending Hostname: yes
    Callhome hostname privacy: DISABLED
    Smart Licensing hostname privacy: DISABLED

  Version privacy: DISABLED
Transport:
  Type: Callhome
Policy:
  Policy in use: Merged from multiple sources.

  Reporting ACK required: yes (CISCO default)
  Unenforced/Non-Export Perpetual Attributes:
    First report requirement (days): 365 (CISCO
 default)
    Reporting frequency (days): 0 (CISCO
default)
   Report on change (days): 90 (CISCO default)

  Unenforced/Non-Export Subscription
Attributes:
    First report requirement (days): 90 (CISCO
 default)
    Reporting frequency (days): 90 (CISCO
default)
   Report on change (days): 90 (CISCO default)

  Enforced (Perpetual/Subscription) License
Attributes:
    First report requirement (days): 0 (CISCO
 default)
    Reporting frequency (days): 0 (CISCO
default)
    Report on change (days): 0 (CISCO default)

  Export (Perpetual/Subscription) License
Attributes:
    First report requirement (days): 0 (CISCO
 default)
    Reporting frequency (days): 0 (CISCO
default)
    Report on change (days): 0 (CISCO default)
Miscellaneous:
  Custom Id: <empty>
Usage Reporting:
  Last ACK received: Sep 22 13:49:38 2020 PST
  Next ACK deadline: Dec 21 12:02:21 2020 PST
  Reporting push interval: 30 days
  Next ACK push check: Sep 22 12:20:34 2020
PST
  Next report push: Oct 22 12:05:43 2020 PST
  Last report push: Sep 22 12:05:43 2020 PST
  Last report file write: <none>
Trust Code Installed:
  Active: PID:C9500-16X,SN:FCW2233A5ZV
  INSTALLED on Sep 22 12:02:20 2020 PST
  Standby: PID:C9500-16X,SN:FCW2233A5ZY
  INSTALLED on Sep 22 12:02:20 2020 PST
```

show license udi（スマートライセンシング）	**show license udi**（スマートライセンシング）
	これは高可用性セットアップであり、このコマンドによってセットアップ内のすべての UDI が表示されます。
`Device# show license udi` `UDI: PID:C9500-16X,SN:FCW2233A5ZV` `HA UDI List:` `Active:PID:C9500-16X,SN:FCW2233A5ZV` `Standby:PID:C9500-16X,SN:FCW2233A5ZY`	`Device# show license udi` `UDI: PID:C9500-16X,SN:FCW2233A5ZV` `HA UDI List:` `Active:PID:C9500-16X,SN:FCW2233A5ZV` `Standby:PID:C9500-16X,SN:FCW2233A5ZY`

移行後の CSSM Web UI

https://software.cisco.com で CSSM Web UI にログインします。[Smart Software Licensing] で、[Manage licenses] リンクをクリックします。

[Inventory] タブをクリックします。[Virtual Account] ドロップダウンリストから、必要なバーチャルアカウントを選択します。[Product Instances] タブをクリックします。

スマートライセンシング環境で登録されたライセンスは、製品インスタンスのホスト名と共に [Name] 列に表示されていました。ポリシーを使用したスマートライセンシングにアップグレードすると、製品インスタンスの UDI と共に表示されるようになります。移行したすべての UDI が表示されます。この例では、PID:C9500-16X,SN:FCW2233A5ZV および PID:C9500-16X,SN:FCW2233A5ZY がこれに該当します。

アクティブな製品インスタンスの使用状況のみが報告されるため、PID:C9500-16X,SN:FCW2233A5ZV の [License Usage] にはライセンス使用情報が表示されます。スタンバイの使用状況は報告されず、スタンバイの [License Usage] セクションには [No Records Found] と表示されます。

常にアクティブの使用状況が報告されるため、この高可用性セットアップのアクティブが変更されると、新しいアクティブな製品インスタンスのライセンス使用情報が表示され、使用状況が報告されるようになります。

図 ***11*** *:* スマートライセンシングからポリシーを使用したスマートライセンシングへ：移行後のアクティブおよびスタンバイ製品インスタンス

図 *12 :* スマートライセンシングからポリシーを使用したスマートライセンシングへ：アクティブな製品インスタンスでの *UDI* とライセンス使用状況

移行後のレポート

製品インスタンスは、ポリシーに基づいて次の RUM レポートを CSSM に送信します。

より頻繁にレポートを作成するようにレポート間隔を変更する場合は、製品インスタンスで **license smart usage interval** コマンドを設定します。シンタックスの詳細については、対応するリリースのコマンドリファレンスで *license smart (global config)* コマンドを参照してください。

例：RTU ライセンシングからポリシーを使用したスマートライセンシングへ

次に、使用権（RTU）ライセンシングからポリシーを使用したスマートライセンシングに移行する Cisco Catalyst 9300 スイッチの例を示します。これはアクティブと他のメンバーを含むセットアップの例です。

RTU ライセンシングは、Cisco IOS XE Fuji 16.8.x までの Cisco Catalyst 9300、9400、および 9500 シリーズ スイッチで使用できます。スマートライセンシングは、Cisco IOS XE Fuji 16.9.1 から導入されました。

ソフトウェアバージョンを、ポリシーを使用したスマートライセンシングをサポートするバージョンにアップグレードすると、すべてのライセンスが IN USE として表示され、`Cisco default` ポリシーが製品インスタンスに適用されます。アドオンライセンスが使用されている場合、`Cisco default` ポリシーでは 90 日間の使用状況レポートが必要です。RTU ライセンスモデルがサポートされていたときに Cisco Catalyst アクセス、コア、およびアグリゲーションスイッチで使用可能な輸出規制ライセンスまたは適用ライセンスはなかったため、どの機能も失われていません。

- 表 15 : RTU ライセンシングからポリシーを使用したスマートライセンシングへ：show コマンド
- 移行後の CSSM Web UI （194 ページ）
- 移行後のレポート （194 ページ）

次の表に、ポリシーを使用したスマートライセンシングへのアップグレード後に、**show** コマンドの出力でチェックすべき主な変更点または新しいフィールドを示します。

表 *15 : RTU* ライセンシングからポリシーを使用したスマートライセンシングへ：*show* コマンド

アップグレード前	アップグレード後
show license right-to-use summary（RTU ライセンシング）	**show license summary**（ポリシーを使用したスマートライセンシング） すべてのライセンスが移行され、`IN USE` になっています。

アップグレード前	アップグレード後
Device# **show license right-to-use summary** License Name Type Period left --- network-essentials Permanent Lifetime dna-essentials Subscription CSSM Managed --- License Level In Use: network-essentials+dna-essentials Subscription License Level on Reboot: network-essentials+dna-essentials Subscription	Device#**show license summary** License Usage: License Entitlement Tag Count Status --- network-essentials (C9300-24 Network Essen...) 2 **IN USE** dna-essentials (C9300-24 DNA Essentials) 2 **IN USE** network-essentials (C9300-48 Network Essen...) 1 **IN USE** dna-essentials (C9300-48 DNA Essentials) 1 **IN USE**

show license right-to-use usage（スマートライセンシング）	**show license usage**（ポリシーを使用したスマートライセンシング） すべてのライセンス（無期限、サブスクリプション）が移行され、それらのライセンスは現在 `IN USE` になっており、タイプには `Perpetual` と `Subscription` があります。 [Enforcement Type] フィールドに NOT ENFORCED と表示されます。これは、アップグレード前に使用されていたすべてのライセンスが不適用ライセンスであったためです。

```
Device# show license right-to-use usage

Slot# License Name Type usage-duration(y:m:d) In-Use
EULA
-------------------------------------------------------
1 network-essentials Permanent 00:00:00 yes yes
1 network-essentials Evaluation 00:00:00 no no
1 network-essentials Subscription 00:00:00 no no
1 network-advantage Permanent 00:00:00 no no
1 network-advantage Evaluation 00:00:00 no no
1 network-advantage Subscription 00:00:00 no no
1 dna-essentials Evaluation 00:00:00 no no
1 dna-essentials Subscription 00:00:00 yes yes
1 dna-advantage Evaluation 00:00:00 no no
1 dna-advantage Subscription 00:00:00 no no
-------------------------------------------------------
Slot# License Name Type usage-duration(y:m:d) In-Use
EULA
-------------------------------------------------------
2 network-essentials Permanent 00:00:00 yes yes
2 network-essentials Evaluation 00:00:00 no no
2 network-essentials Subscription 00:00:00 no no
2 network-advantage Permanent 00:00:00 no no
2 network-advantage Evaluation 00:00:00 no no
2 network-advantage Subscription 00:00:00 no no
2 dna-essentials Evaluation 00:00:00 no no
2 dna-essentials Subscription 00:00:00 yes yes
2 dna-advantage Evaluation 00:00:00 no no
2 dna-advantage Subscription 00:00:00 no no
-------------------------------------------------------
Slot# License Name Type usage-duration(y:m:d) In-Use
EULA
-------------------------------------------------------
3 network-essentials Permanent 00:00:00 yes yes
3 network-essentials Evaluation 00:00:00 no no
3 network-essentials Subscription 00:00:00 no no
3 network-advantage Permanent 00:00:00 no no
3 network-advantage Evaluation 00:00:00 no no
3 network-advantage Subscription 00:00:00 no no
3 dna-essentials Evaluation 00:00:00 no no
3 dna-essentials Subscription 00:00:00 yes yes
3 dna-advantage Evaluation 00:00:00 no no
3 dna-advantage Subscription 00:00:00 no no
-------------------------------------------------------
```

```
Device# show license usage

License Authorization:
  Status: Not Applicable
network-advantage (C9300-24 Network Advantage):
  Description: C9300-24 Network Advantage
  Count: 2
  Version: 1.0
  Status: IN USE
  Export status: NOT RESTRICTED
  Feature Name: network-advantage
  Feature Description: C9300-24 Network Advantage
  Enforcement type: NOT ENFORCED
  License type: Perpetual
dna-advantage (C9300-24 DNA Advantage):
  Description: C9300-24 DNA Advantage
  Count: 2
  Version: 1.0
  Status: IN USE
  Export status: NOT RESTRICTED
  Feature Name: dna-advantage
  Feature Description: C9300-24 DNA Advantage
  Enforcement type: NOT ENFORCED
  License type: Subscription
network-advantage (C9300-48 Network Advantage):
  Description: C9300-48 Network Advantage
  Count: 1
  Version: 1.0
  Status: IN USE
  Export status: NOT RESTRICTED
  Feature Name: network-advantage
  Feature Description: C9300-48 Network Advantage
  Enforcement type: NOT ENFORCED
  License type: Perpetual
dna-advantage (C9300-48 DNA Advantage):
  Description: C9300-48 DNA Advantage
  Count: 1
  Version: 1.0
  Status: IN USE
  Export status: NOT RESTRICTED
  Feature Name: dna-advantage
  Feature Description: C9300-48 DNA Advantage
  Enforcement type: NOT ENFORCED
  License type: Subscription
```

show license right-to-use（RTU ライセンシング）

show license status（ポリシーを使用したスマートライセンシング）

`Transport:` フィールドにオフになっていることが表示されます。

`Trust Code Installed:` フィールドには、信頼コードがインストールされていないことが表示されます。

`Usage Reporting:` ヘッダーの `Next report push:` フィールドには、次の RUM レポートを CSSM に送信するタイミングに関する情報が表示されます。

```
Device# show license right-to-use
Slot# License Name Type Period left
--------------------------------------------------
1 network-essentials Permanent Lifetime
1 dna-essentials Subscription CSSM Managed
--------------------------------------------------
License Level on Reboot:
network-essentials+dna-essentials
Subscription

Slot# License Name Type Period left
--------------------------------------------------
2 network-essentials Permanent Lifetime
2 dna-essentials Subscription CSSM Managed
--------------------------------------------------
License Level on Reboot:
network-essentials+dna-essentials
Subscription

Slot# License Name Type Period left
--------------------------------------------------
3 network-essentials Permanent Lifetime
3 dna-essentials Subscription CSSM Managed
--------------------------------------------------
License Level on Reboot:
network-essentials+dna-essentials
Subscription
```

```
Device# show license status
Utility:
  Status: DISABLED
Smart Licensing Using Policy:
  Status: ENABLED
Data Privacy:
  Sending Hostname: yes
    Callhome hostname privacy: DISABLED
    Smart Licensing hostname privacy: DISABLED
  Version privacy: DISABLED
Transport:
  Type: Transport Off
Policy:
  Policy in use: Merged from multiple sources.
  Reporting ACK required: yes (CISCO default)
  Unenforced/Non-Export Perpetual Attributes:
    First report requirement (days): 365 (CISCO default)

    Reporting frequency (days): 0 (CISCO default)
    Report on change (days): 90 (CISCO default)
  Unenforced/Non-Export Subscription Attributes:
    First report requirement (days): 90 (CISCO default)

    Reporting frequency (days): 90 (CISCO default)
    Report on change (days): 90 (CISCO default)
  Enforced (Perpetual/Subscription) License Attributes:

    First report requirement (days): 0 (CISCO default)
    Reporting frequency (days): 0 (CISCO default)
    Report on change (days): 0 (CISCO default)
  Export (Perpetual/Subscription) License Attributes:
    First report requirement (days): 0 (CISCO default)
    Reporting frequency (days): 0 (CISCO default)
    Report on change (days): 0 (CISCO default)
Miscellaneous:
  Custom Id: <empty>
Usage Reporting:
  Last ACK received: <none>
  Next ACK deadline: Jan 26 10:27:59 2021 PST
  Reporting push interval: 20  days
  Next ACK push check: <none>
  Next report push: Oct 28 10:29:59 2020 PST
  Last report push: <none>
  Last report file write: <none>
Trust Code Installed: <none>
```

移行後の CSSM Web UI

CSSM Web UI に変更はありません。

移行後のレポート

サポートされているトポロジのいずれかを実装し、レポート要件に適合するようにします。サポートされるトポロジ（147ページ）およびポリシーを使用したスマートライセンシングの設定方法：トポロジ別のワークフロー（164ページ）を参照してください。使用可能なレポートメソッドは、実装するトポロジによって異なります。

例：SLR からポリシーを使用したスマートライセンシングへ

次に、特定のライセンス予約（SLR）からポリシーを使用したスマートライセンシングに移行する Cisco Catalyst 9500 スイッチの例を示します。これはアクティブとスタンバイを含む高可用性セットアップの例です。

ライセンスの変換は自動的に行われ、承認コードが移行されます。移行を完了するためにこれ以上の操作は必要ありません。移行後はCSSM への接続なし、CSLU なし（152 ページ）トポロジが有効になります。ポリシーを使用したスマートライセンシング環境の SLR 承認コードについては、承認コード（143 ページ）を参照してください。

- 表 16 : SLR からポリシーを使用したスマートライセンシングへ：show コマンド
- 移行後の CSSM Web UI（201 ページ）
- 移行後のレポート（204 ページ）

show コマンドは、移行の前後に確認すべき以下の重要なフィールドを抽出して出力します。

表 *16 : SLR* からポリシーを使用したスマートライセンシングへ：*show* コマンド

アップグレード前	アップグレード後
show license summary（SLR） `Registration` ステータスフィールドと `License Authorization` ステータスフィールドに、ライセンスについて `REGISTERED - SPECIFIC LICENSE RESERVATION` および `AUTHORIZED - RESERVED` と表示されます。 `Device# show license summary` `Smart Licensing is ENABLED` `License Reservation is ENABLED` `Registration:` `  Status: REGISTERED - SPECIFIC LICENSE RESERVATION` `Export-Controlled Functionality: ALLOWED` `License Authorization:` `  Status: AUTHORIZED - RESERVED` `License Usage:` `License  Entitlement tag  Count Status` `---` `C9500 Network Advantage(C9500 Network Advantage)  2 AUTHORIZED` `C9500-DNA-16X-A  (C9500-16X DNA Advantage)  2 AUTHORIZED`	**show license summary**（ポリシーを使用したスマートライセンシング） `Status` フィールドに、ライセンスについて、登録済みおよび承認済みではなく `IN USE` と表示されます。 `Device# show license summary` `License Reservation is ENABLED` `License Usage:` `License  Entitlement tag  Count Status` `---` `network-advantage(C9500 Network Advantage)  2 IN USE` `dna-advantage  (C9500-16X DNA Advantage)  2 IN USE`

show license reservation（SLR）	**show license all**（ポリシーを使用したスマートライセンシング） `License Authorizations` ヘッダー：アクティブおよびスタンバイ製品インスタンスのベース（C9500 Network Advantage）ライセンスおよびアドオン（C9500-DNA-16X-A）ライセンスが特定のライセンス予約で承認されたことを示します。`Authorization type:` フィールドに SPECIFIC INSTALLED と表示されます。 `Last Confirmation code:` フィールド：高可用性セットアップのアクティブおよびスタンバイ製品インスタンスの SLR 承認コードが正常に移行されたことを示します。

```
Device# show license reservation
License reservation: ENABLED
Overall status:
  Active: PID:C9500-16X,SN:FCW2233A5ZV
      Reservation status: SPECIFIC INSTALLED on Aug 31
 10:15:01 2020 PDT
      Export-Controlled Functionality: ALLOWED
      Last Confirmation code: 4bfbea7f
  Standby: PID:C9500-16X,SN:FCW2233A5ZY
      Reservation status: SPECIFIC INSTALLED on Aug 31
 10:15:01 2020 PDT
      Export-Controlled Functionality: ALLOWED
      Last Confirmation code: 9394f196
Specified license reservations:
  C9500 Network Advantage (C9500 Network Advantage):
    Description: C9500 Network Advantage
    Total reserved count: 2
    Term information:
      Active: PID:C9500-16X,SN:FCW2233A5ZV
        License type: PERPETUAL
          Term Count: 1
      Standby: PID:C9500-16X,SN:FCW2233A5ZY
        License type: PERPETUAL
          Term Count: 1
  C9500-DNA-16X-A (C9500-16X DNA Advantage):
    Description: C9500-DNA-16X-A
    Total reserved count: 2
    Term information:
      Active: PID:C9500-16X,SN:FCW2233A5ZV
        License type: TERM
          Start Date: 2020-MAR-17 UTC
          End Date: 2021-MAR-17 UTC
          Term Count: 1
      Standby: PID:C9500-16X,SN:FCW2233A5ZY
```

```
Device# show license reservation

Smart Licensing Status
======================
Smart Licensing is ENABLED
License Reservation is ENABLED
Export Authorization Key:
  Features Authorized:
    <none>
Utility:
  Status: DISABLED
Smart Licensing Using Policy:
  Status: ENABLED
Data Privacy:
  Sending Hostname: yes
    Callhome hostname privacy: DISABLED
    Smart Licensing hostname privacy: DISABLED
  Version privacy: DISABLED
Transport:
  Type: Transport Off
Miscellaneous:
  Custom Id: <empty>
Policy:
  Policy in use: Merged from multiple sources.
  Reporting ACK required: yes (CISCO default)
  Unenforced/Non-Export Perpetual Attributes:
    First report requirement (days): 365 (CISCO default)

    Reporting frequency (days): 0 (CISCO default)
    Report on change (days): 90 (CISCO default)
  Unenforced/Non-Export Subscription Attributes:
    First report requirement (days): 90 (CISCO default)

    Reporting frequency (days): 90 (CISCO default)
    Report on change (days): 90 (CISCO default)
  Enforced (Perpetual/Subscription) License Attributes:

    First report requirement (days): 0 (CISCO default)
    Reporting frequency (days): 0 (CISCO default)
    Report on change (days): 0 (CISCO default)
  Export (Perpetual/Subscription) License Attributes:
    First report requirement (days): 0 (CISCO default)
    Reporting frequency (days): 0 (CISCO default)
    Report on change (days): 0 (CISCO default)
Usage Reporting:
  Last ACK received: <none>
  Next ACK deadline: Nov 29 10:50:05 2020 PDT
  Reporting Interval: 30
  Next ACK push check: <none>
  Next report push: Aug 31 10:52:05 2020 PDT
  Last report push: <none>
  Last report file write: <none>
Trust Code Installed: <none>
License Usage
=============
network-advantage (C9500 Network Advantage):
  Description: network-advantage
  Count: 2
  Version: 1.0
  Status: IN USE
  Export status: NOT RESTRICTED
  Feature Name: network-advantage
  Feature Description: network-advantage
  Enforcement type: NOT ENFORCED
```

```
  License type: Perpetual
  Reservation:
    Reservation status: SPECIFIC INSTALLED
    Total reserved count: 2
dna-advantage (C9500-16X DNA Advantage):
  Description: C9500-16X DNA Advantage
  Count: 2
  Version: 1.0
  Status: IN USE
  Export status: NOT RESTRICTED
  Feature Name: dna-advantage
  Feature Description: C9500-16X DNA Advantage
  Enforcement type: NOT ENFORCED
  License type: Subscription
  Reservation:
    Reservation status: SPECIFIC INSTALLED
    Total reserved count: 2
Product Information
===================
UDI: PID:C9500-16X,SN:FCW2233A5ZV
HA UDI List:
    Active:PID:C9500-16X,SN:FCW2233A5ZV
    Standby:PID:C9500-16X,SN:FCW2233A5ZY
Agent Version
=============
Smart Agent for Licensing: 5.0.5_rel/42
License Authorizations
======================
Overall status:
  Active: PID:C9500-16X,SN:FCW2233A5ZV
     Status: SPECIFIC INSTALLED on Aug 31 10:15:01 2020
 PDT
      Last Confirmation code: 4bfbea7f
  Standby: PID:C9500-16X,SN:FCW2233A5ZY
     Status: SPECIFIC INSTALLED on Aug 31 10:15:01 2020
 PDT
      Last Confirmation code: 9394f196
Specified license reservations:
  C9500 Network Advantage (C9500 Network Advantage):
    Description: C9500 Network Advantage
    Total reserved count: 2
    Enforcement type: NOT ENFORCED
    Term information:
      Active: PID:C9500-16X,SN:FCW2233A5ZV
        Authorization type: SPECIFIC INSTALLED on Aug
31 10:15:01 2020 PDT
          License type: PERPETUAL
            Term Count: 1
      Standby: PID:C9500-16X,SN:FCW2233A5ZY
        Authorization type: SPECIFIC INSTALLED on Aug
31 10:15:01 2020 PDT
          License type: PERPETUAL
            Term Count: 1
  C9500-DNA-16X-A (C9500-16X DNA Advantage):
    Description: C9500-DNA-16X-A
    Total reserved count: 2
    Enforcement type: NOT ENFORCED
    Term information:
      Active: PID:C9500-16X,SN:FCW2233A5ZV
        Authorization type: SPECIFIC INSTALLED on Aug
31 10:15:01 2020 PDT
          License type: PERPETUAL
            Term Count: 1
      Standby: PID:C9500-16X,SN:FCW2233A5ZY
```

	`Authorization type: SPECIFIC INSTALLED on Aug 31 10:15:01 2020 PDT` `License type: PERPETUAL` `Term Count: 1` `Purchased Licenses:` `No Purchase Information Available` `Derived Licenses:` `Entitlement Tag:` `regid.2017-03.com.cisco.advantagek9-Nyquist-C9500, 1.0_f1563759-2e03-4a4c-bec5-5feec525a12c` `Entitlement Tag:` `regid.2017-07.com.cisco.C9500-DNA-16X-A, 1.0_ef3574d1-156b-486a-864f-9f779ff3ee49`
show license status（SLR）	**show license status**（ポリシーを使用したスマートライセンシング） `Transport:` ヘッダー：`Type:` は、転送タイプがオフに設定されていることを示します。 `Usage Reporting:` ヘッダー：`Next report push:` フィールドは、次の RUM レポートを CSSM にアップロードする必要があるかどうか、およびアップロードする必要があるのはいつかを示します。

```
Device# show license status

Smart Licensing is ENABLED
Utility:
  Status: DISABLED
License Reservation is ENABLED
Data Privacy:
  Sending Hostname: yes
    Callhome hostname privacy: DISABLED
    Smart Licensing hostname privacy: DISABLED
  Version privacy: DISABLED
Transport:
  Type: Callhome
Registration:
  Status: REGISTERED - SPECIFIC LICENSE RESERVATION
  Export-Controlled Functionality: ALLOWED
  Initial Registration: SUCCEEDED on Aug 31 11:07:39
2020 PDT
License Authorization:
  Status: AUTHORIZED - RESERVED on Aug 31 10:15:01 2020
 PDT
Export Authorization Key:
  Features Authorized:
    <none>
        License type: TERM
          Start Date: 2020-MAR-17 UTC
          End Date: 2021-MAR-17 UTC
          Term Count: 1
```

```
Device# show license status

Utility:
  Status: DISABLED
License Reservation is ENABLED
Data Privacy:
  Sending Hostname: yes
    Callhome hostname privacy: DISABLED
    Smart Licensing hostname privacy: DISABLED
   Version privacy: DISABLED
Transport:
  Type: Transport Off
Policy:
  Policy in use: Merged from multiple sources.
  Reporting ACK required: yes (CISCO default)
  Unenforced/Non-Export Perpetual Attributes:
    First report requirement (days): 365 (CISCO default)

    Reporting frequency (days): 0 (CISCO default)
    Report on change (days): 90 (CISCO default)
  Unenforced/Non-Export Subscription Attributes:
    First report requirement (days): 90 (CISCO default)

    Reporting frequency (days): 90 (CISCO default)
    Report on change (days): 90 (CISCO default)
  Enforced (Perpetual/Subscription) License Attributes:

    First report requirement (days): 0 (CISCO default)
    Reporting frequency (days): 0 (CISCO default)
    Report on change (days): 0 (CISCO default)
  Export (Perpetual/Subscription) License Attributes:
    First report requirement (days): 0 (CISCO default)
    Reporting frequency (days): 0 (CISCO default)
    Report on change (days): 0 (CISCO default)
Miscellaneous:
  Custom Id: <empty>
Usage Reporting:
  Last ACK received: <none>
  Next ACK deadline: Nov 29 10:50:05 2020 PDT
  Reporting Interval: 30
  Next ACK push check: <none>
  Next report push: Aug 31 10:52:05 2020 PDT
  Last report push: <none>
  Last report file write: <none>
Trust Code Installed: <none>
```

移行後の CSSM Web UI

CSSM では、[Product Instances] タブに変更はありません。使用状況レポートがまだないため、[Last Contact] 列には「Reserved Licenses」と表示されます。

必要な RUM レポートがアップロードされ、「Reserved Licenses（予約済みライセンス）」が確認されると、ライセンスの使用状況がアクティブな PID 製品インスタンスのみで表示されるようになります。

図 *13: SLR* からポリシーを使用したスマートライセンシングへ：移行後、レポート前のアクティブおよびスタンバイ製品インスタンス

図 *14: SLR* からポリシーを使用したスマートライセンシングへ：移行後、レポート後のアクティブおよびスタンバイ製品インスタンス

移行後のレポート

SLR ライセンスは、ライセンスの使用状況が変化した場合にのみレポートを必要とします（たとえば、アドオンライセンスを指定された期間使用する場合）。ポリシー（**show license status**）によって変化が示されるか、変化に関する syslog メッセージが発信されます。

製品インスタンスとのすべての通信を無効にしているため、ライセンスの使用状況をレポートするには、RUM レポートをファイルに保存してから、CSSM にアップロードする必要があります（インターネットとシスコに接続されているワークステーションからアップロード）。

1. RUM レポートの生成と保存

   license smart save usage コマンドを特権 EXEC モードで入力します。次の例では、すべての RUM レポートがファイル `all_rum.txt` で製品インスタンスのフラッシュメモリに保存されます。シンタックスの詳細については、対応するリリースのコマンドリファレンスで *license smart (privileged EXEC)* コマンドを参照してください。この例では、ファイルはまずブートフラッシュに保存され、次に TFTP の場所にコピーされます。

   ```
   Device# license smart save usage all bootflash:all_rum.txt
   Device# copy bootflash:all_rum.txt tftp://10.8.0.6/all_rum.txt
   ```

2. 使用状況データを CSSM にアップロード： CSSM への使用状況データのアップロードと ACK のダウンロード （260 ページ）

3. ACK を製品インスタンスにインストール： 製品インスタンスへのファイルのインストール （261 ページ）

例：評価ライセンスまたは期限切れライセンスからポリシーを使用したスマートライセンシングへ

以下は、評価ライセンス（スマートライセンシング）を、ポリシーを使用したスマートライセンシングに移行した Cisco Catalyst 9500 スイッチの例です。

評価ライセンスの概念は、ポリシーを使用したスマートライセンスには適用されません。ソフトウェアバージョンを、ポリシーを使用したスマートライセンシングをサポートするバージョンにアップグレードすると、すべてのライセンスが IN USE として表示され、シスコのデフォルトポリシーが製品インスタンスに適用されます。以前のライセンスモデルが有効であったときに Cisco Catalyst アクセス、コア、およびアグリゲーションスイッチで使用可能なエクスポート制御されたライセンスまたは適用されたライセンスはなかったため、どの機能も失われていません。

- 表 17 : 評価ライセンスまたは期限切れライセンスからポリシーを使用したスマートライセンシングへ：show コマンド
- 移行後の CSSM Web UI （207 ページ）
- 移行後のレポート （207 ページ）

次の表に、ポリシーを使用したスマートライセンシングへのアップグレード後に、**show** コマンドの出力でチェックすべき主な変更点または新しいフィールドを示します。

表 *17*：評価ライセンスまたは期限切れライセンスからポリシーを使用したスマートライセンシングへ：*show* コマンド

アップグレード前	アップグレード後
show license summary（スマートライセンシング、評価モード） ライセンスは UNREGISTERED で、EVAL MODE になっています。 `Device# show license summary` `Smart Licensing is ENABLED` `Registration:` `Status: UNREGISTERED` `Export-Controlled Functionality: NOT ALLOWED` `License Authorization:` `Status: EVAL MODE` `Evaluation Period Remaining: 89 days, 21 hours, 37 minutes,` `30 seconds` `License Usage:` `License Entitlement tag Count Status` `---` `(C9500 Network Advantage) 2 EVAL MODE` `(C9500-16X DNA Advantage) 2 EVAL MODE`	**show license summary**（ポリシーを使用したスマートライセンシング） すべてのライセンスが移行され、IN USE になっています。評価モードライセンスがありません。 `Device# show license summary` `License Usage:` `License Entitlement tag Count Status` `---` `network-advantage (C9500 Network Advantage) 2 IN USE` `dna-advantage (C9500-16X DNA Advantage) 2 IN USE`
show license usage（スマートライセンシング、評価モード） `Device# show license usage` `License Authorization:` `Status: EVAL MODE` `Evaluation Period Remaining: 89 days, 21 hours, 37 minutes,` `21 seconds` `(C9500 Network Advantage):` `Description:` `Count: 2` `Version: 1.0` `Status: EVAL MODE` `Export status: NOT RESTRICTED` `(C9500-16X DNA Advantage):` `Description:` `Count: 2` `Version: 1.0` `Status: EVAL MODE` `Export status: NOT RESTRICTED`	**show license usage**（ポリシーを使用したスマートライセンシング） [Enforcement Type] フィールドに NOT ENFORCED と表示されます。これは、アップグレード前に使用されていたすべてのライセンスが適用されていないためです。 `Device# show license usage` `License Authorization:` `Status: Not Applicable` `network-advantage (C9500 Network Advantage):` `Description: network-advantage` `Count: 2` `Version: 1.0` `Status: IN USE` `Export status: NOT RESTRICTED` `Feature Name: network-advantage` `Feature Description: network-advantage` `Enforcement type: NOT ENFORCED` `License type: Perpetual` `dna-advantage (C9500-16X DNA Advantage):` `Description: C9500-16X DNA Advantage` `Count: 2` `Version: 1.0` `Status: IN USE` `Export status: NOT RESTRICTED` `Feature Name: dna-advantage` `Feature Description: C9500-16X DNA Advantage` `Enforcement type: NOT ENFORCED` `License type: Subscription`

show license status（スマートライセンス、評価モード）	**show license status**（ポリシーを使用したスマートライセンシング） `Transport:` フィールドにオフになっていることが表示されます。 `Policy` フィールドには、シスコのデフォルトポリシーが適用されていることが示されます。 `Trust Code Installed:` フィールドには、信頼コードがインストールされていないことが表示されます。 `Usage Reporting:` ヘッダー：`Next report push:` フィールドには、次の RUM レポートを CSSM に送信するタイミングに関する情報が表示されます。

```
Switch# show license status

Smart Licensing is ENABLED
Utility:
Status: DISABLED
Data Privacy:
Sending Hostname: yes
Callhome hostname privacy: DISABLED
Smart Licensing hostname privacy: DISABLED
Version privacy: DISABLED
Transport:
Type: Callhome
Registration:
Status: UNREGISTERED
Export-Controlled Functionality: NOT ALLOWED
License Authorization:
Status: EVAL MODE
Evaluation Period Remaining: 89 days, 21 hours, 37
minutes, 15 seconds
Export Authorization Key:
Features Authorized:
<none>
Miscellaneus:
Custom Id: <empty>
```

```
Switch# show license status

Utility:
  Status: DISABLED
Smart Licensing Using Policy:
  Status: ENABLED
Data Privacy:
  Sending Hostname: yes
    Callhome hostname privacy: DISABLED
    Smart Licensing hostname privacy: DISABLED
  Version privacy: DISABLED
Transport:
  Type: Transport Off
Policy:
  Policy in use: Merged from multiple sources.
  Reporting ACK required: yes (CISCO default)
  Unenforced/Non-Export Perpetual Attributes:
    First report requirement (days): 365 (CISCO default)

    Reporting frequency (days): 0 (CISCO default)
    Report on change (days): 90 (CISCO default)
  Unenforced/Non-Export Subscription Attributes:
    First report requirement (days): 90 (CISCO default)

    Reporting frequency (days): 90 (CISCO default)
    Report on change (days): 90 (CISCO default)
  Enforced (Perpetual/Subscription) License Attributes:

    First report requirement (days): 0 (CISCO default)
    Reporting frequency (days): 0 (CISCO default)
    Report on change (days): 0 (CISCO default)
  Export (Perpetual/Subscription) License Attributes:
    First report requirement (days): 0 (CISCO default)
    Reporting frequency (days): 0 (CISCO default)
    Report on change (days): 0 (CISCO default)
Miscellaneous:
  Custom Id: <empty>
Usage Reporting:
  Last ACK received: <none>
  Next ACK deadline: Jan 26 10:27:59 2021 PST
  Reporting push interval: 20  days
  Next ACK push check: <none>
  Next report push: Oct 28 10:29:59 2020 PST
  Last report push: <none>
  Last report file write: <none>
Trust Code Installed: <none>
```

移行後の CSSM Web UI

CSSM Web UI に変更はありません。

移行後のレポート

サポートされているトポロジのいずれかを実装し、レポート要件に適合するようにします。サポートされるトポロジ（147ページ）およびポリシーを使用したスマートライセンシングの設定方法：トポロジ別のワークフロー（164ページ）を参照してください。使用可能なレポートメソッドは、実装するトポロジによって異なります。

Smart Licensing Using Policy をサポートする SSM オンプレミスのバージョンへの移行

必要な最小バージョンよりも前の SSM オンプレミス のバージョンを使用している場合（SSM オンプレミス （142 ページ）を参照）、SSMオンプレミスのバージョン、製品インスタンスを移行するために従う必要があるプロセスや手順、および該当する場合はSLACのインストールのような他のタスクの概要として使用してください。

1. SSM オンプレミスをアップグレードします。

   必要な最小バージョンであるバージョン 8、リリース 202102 以降にアップグレードします。

   『Cisco Smart Software Manager On-Prem Migration Guide』を参照してください。

2. 製品インスタンスをアップグレードします。

   サポートされている製品インスタンスに Smart Licensing Using Policy が導入された時期については、サポート対象製品 （139 ページ）を参照してください。

   アップグレード手順については、スイッチ ソフトウェアのアップグレード （183 ページ）を参照してください。

3. CSSM へのローカルアカウントの再登録

   オンラインとオフラインのオプションを使用できます。『Cisco Smart Software Manager On-Prem Migration Guide』の「*Re-Registering a local Account (Online Mode)*」または「*Manually Re-Registering a Local Account (Offline Mode)*」を参照してください。

   再登録が完了すると、次のイベントが自動的に発生します。

   - SSMオンプレミスは、SSMオンプレミスのテナントを指す新しいトランスポートURLで応答します。
   - 製品インスタンスのトランスポートタイプ設定が **call-home** または **smart** から **cslu** に変更されます。トランスポート URL も自動的に更新されます。

4. 特権 EXEC モードで **copy running-config startup-config** コマンドを入力して、製品インスタンスの設定変更を保存します。

5. 製品インスタンスの古いオンプレミス スマート ライセンス証明書をクリアし、製品インスタンスをリロードします。この後は設定変更を保存しないでください。

（注） この手順は、製品インスタンスで実行されているソフトウェアバージョンが Cisco IOS XE Amsterdam 17.3.x または Cisco IOS XE Bengaluru 17.4.x の場合にのみ必要です。

特権 EXEC モードで **licence smart factory reset** コマンドと **reload** コマンドを入力します。

```
Device# licence smart factory reset
Device# reload
```

6. 使用状況の同期の実行

   1. 製品インスタンスに特権 EXEC モードで **license smart sync** {**all**|**local**} コマンドを入力します。これにより、SSM オンプレミスと製品インスタンスが同期され、保留中のデータが送受信されます。

      ```
      Device(config)# license smart sync local
      ```

      これは、SSM オンプレミス UI で確認できます。[Inventory] > [SL Using Policy] に移動します。[Alerts] 列に、「Usage report from product instance」というメッセージが表示されます。

   2. 使用状況情報を CSSM と同期します（いずれかを選択）。

      - オプション 1：

        SSM オンプレミスが CSSM に接続されている場合：SSM オンプレミス UI の [Smart Licensing] ワークスペースで、[Reports] > [Usage Schedules] > [Synchronize now with Cisco] に移動します。

      - オプション 2：

        SSM オンプレミスが CSSM に接続されていません。使用状況データのエクスポートとインポート（SSM オンプレミス UI）（235 ページ）を参照してください。

結果：

移行および使用状況の最初の同期が完了しました。製品インスタンスとライセンス使用状況情報が SSM オンプレミスに表示されるようになりました。

後続のレポートには、次のオプションが含まれています。

- 製品インスタンスと SSM オンプレミスとの間でデータを同期するには、次の手順を実行します。
  - レポート間隔を設定して、製品スタンスと SSM オンプレミスとの間の定期的な同期をスケジュールします。グローバル コンフィギュレーション モードで **license smart usage interval** *interval_in_days* コマンドを入力します。

    製品インスタンスが次にいつ RUM レポートを送信するかを確認するには、特権 EXEC モードで **show license all** コマンドを入力し、出力の [Next report push:] フィールドを確認します。

  - 製品インスタンスと SSM オンプレミスとの間でアドホックまたはオンデマンドの同期を行うには、**license smart sync** 特権 EXEC コマンドを入力します。

- 使用状況情報を CSSM と同期するには、次の手順を実行します。
  - CSSM との定期的な同期をスケジュールします。SSM オンプレミス UI で、[Reports] > [Usage Schedules] > [Synchronization schedule with Cisco] に移動します。次の頻度情報を入力し、保存します。

- [Days]：同期が実行される頻度を示します。たとえば、2 を入力すると、同期は 2 日に 1 回行われます。
- [Time of Day]：24 時間表記法で、同期が実行される時刻を示します。たとえば、14 hours と 0 minutes を入力すると、ローカルタイムゾーンの午後 2 時（1400）に同期が行われます。

- レポートに必要なファイルのアップロードとダウンロードを実行します（使用状況データのエクスポートとインポート（SSM オンプレミス UI）（235 ページ））。

ポリシーを使用したスマートライセンシングのタスクライブラリ

このセクションでは、ポリシーを使用したスマートライセンシングに適用されるタスクのグループ化について説明します。製品インスタンス、CSLU インターフェイス、および CSSM Web UI で実行されるタスクが含まれます。

特定のトポロジを実装するには、対応するワークフローを参照して、適用されるタスクの順序を確認します。ポリシーを使用したスマートライセンシングの設定方法：トポロジ別のワークフロー（164 ページ） を参照してください

追加の設定タスクを実行する場合（たとえば別のライセンスの設定、アドオンライセンスの使用、またはより短いレポート間隔の設定）は、対応するタスクを参照してください。続行する前に、入手可能な場合には「サポートされるトポロジ」を確認してください。

シスコへのログイン（CSLU インターフェイス）

必要に応じて、CSLU で作業するときに接続モードまたは切断モードのいずれかにすることができます。接続モードで作業するには、次の手順を実行してシスコに接続します。

手順

ステップ 1 CSLU のメイン画面で、[Login to Cisco]（画面の右上隅）をクリックします。

ステップ 2 [CCO User Name] と [CCO Password] を入力します。

ステップ 3 CSLU の [Preferences] タブで、シスコ接続トグルに「Cisco Is Available」と表示されていることを確認します。

スマートアカウントとバーチャルアカウントの設定（CSLU インターフェイス）

スマートアカウントとバーチャルアカウントはどちらも [Preferences] タブで設定します。シスコに接続するためのスマートアカウントとバーチャルアカウントの両方を設定するには、次の手順を実行します。

手順

ステップ 1 CSLU のホーム画面から [Preferences] タブを選択します。

ステップ 2 スマートアカウントとバーチャルアカウントの両方を追加するには、次の手順を実行します。

a) [Preferences] 画面で、[Smart Account] フィールドに移動し、[Smart Account Name] を追加します。

b) 次に、[Virtual Account] フィールドに移動し、[Virtual Account Name] を追加します。

CSSM に接続している場合（[Preferences] タブに「Cisco is Available」）、使用可能な SA/VA のリストから選択できます。

CSSM に接続していない場合（[Preferences] タブに「Cisco Is Not Available」）、SA/VA を手動で入力します。

（注） SA/VA 名では大文字と小文字が区別されます。

ステップ 3 [Save] をクリックします。SA/VA アカウントがシステムに保存されます。

一度に 1 つの SA/VA ペアのみが CSLU に存在できます。複数のアカウントを追加することはできません。別の SA/VA ペアに変更するには、ステップ 2a および 2b を繰り返してから [Save] をクリックします。新しい SA/VA アカウントペアは、以前に保存されたペアを置き換えます。

CSLU での製品開始型製品インスタンスの追加（CSLU インターフェイス）

[Preferences] タブを使用してデバイス作成の製品インスタンスを追加するには、次の手順を実行します。

手順

ステップ 1 [Preferences] タブをクリックします。

ステップ 2 [Preferences] 画面で、[Validate Instance] チェックボックスをオフにします。

ステップ 3 [Default Instance Method] を [Product Instance Initiated] に設定し、[Save] をクリックします。

製品インスタンス開始型通信のネットワーク到達可能性の確認

このタスクでは、製品インスタンス開始型通信のネットワーク到達可能性を確認するために必要になる可能性のある設定を提供します。「（必須）」と付いている手順は、すべての製品インスタンスで必須です。他のすべての手順は、製品インスタンスの種類とネットワーク要件に応じて、必須の場合も任意の場合もあります。該当するコマンドを設定します。

始める前に

サポートされるトポロジ：CSLU を介して CSSM に接続（製品インスタンス開始型通信）。

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 モードを有効にします。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	**interface** *interface-type-number* 例： `Device (config)# interface gigabitethernet0/0`	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、VRF に関連付けられたイーサネットインターフェイス、サブインターフェイス、または VLAN を指定します。
ステップ 4	**vrf forwarding** *vrf-name* 例： `Device(config-if)# vrf forwarding Mgmt-vrf`	VRF をレイヤ 3 インターフェイスに対応付けます。このコマンドは、インターフェイスでマルチプロトコル VRF をアクティブにします。
ステップ 5	**ip address** *ip-address mask* 例： `Device(config-if)# ip address 192.168.0.1 255.255.0.0`	VRF の IP アドレスを定義します。
ステップ 6	**negotiation auto** 例： `Device(config-if)# negotiation auto`	インターフェイスの速度およびデュプレックスパラメータの自動ネゴシエーション動作を有効にします。

	コマンドまたはアクション	目的
ステップ 7	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーションモードを終了し、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 8	**ip http client source-interface** *interface-type-number* 例： `Device(config)# ip http client source-interface gigabitethernet0/0`	HTTP クライアントのソース インターフェイスを設定します。
ステップ 9	**ip route** *ip-address ip-mask subnet mask* 例： `Device(config)# ip route vrf mgmt-vrf 192.168.0.1 255.255.0.0 192.168.255.1`	（必須）製品インスタンスにルートとゲートウェイを設定します。スタティックルートまたはダイナミックルートのいずれかを設定できます。
ステップ 10	{**ip**|**ipv6**}**name-server** *server-address 1 ...server-address 6*] 例： `Device(config)# Device(config)# ip name-server vrf mgmt-vrf 173.37.137.85`	VRF インターフェイスでドメインネームシステム（DNS）を設定します。
ステップ 11	**ip domain lookup source-interface** *interface-type-number* 例： `Device(config)# ip domain lookup source-interface gigabitethernet0/0`	DNS ドメインルックアップ用のソースインターフェイスを設定します。
ステップ 12	**ip domain name** *domain-name* 例： `Device(config)# ip domain name example.com`	ドメインの DNS ディスカバリを設定します。ここの例では、ネームサーバーはエントリ `cslu-local.example.com` を作成します。

CSLU での CSLU 開始型製品インスタンスの追加（CSLU インターフェイス）

CSLU インターフェイスを使用して、接続方法を CSLU 開始型に設定できます。この接続方法（モード）により、CSLU は製品インスタンス情報を取得できます。

（注） デフォルトの接続方法は、[Preferences] タブで設定されます。

[Inventory] タブから製品インスタンスを追加するには、次の手順を実行します。

手順

ステップ 1 [Inventory] タブに移動し、[Product Instances] テーブルから [Add Single Product] を選択します。

ステップ 2 [Host] に入力します（ホストの IP アドレス）。

ステップ 3 [Connect Method] を選択し、適切な [CSLU Initiated] 接続方法を選択します。

ステップ 4 右側のパネルで、[Product Instance Login Credentials] をクリックします。画面の左側のパネルが変化して [User Name] フィールドと [Password] フィールドに変わります。

ステップ 5 製品インスタンスの [User Name] と [Password] を入力します。

ステップ 6 [保存（**Save**）] をクリックします。

情報がシステムに保存され、デバイスが [Product Instances] テーブルにリストされて、[Last Contact] には [never] と表示されます。

使用状況レポートの収集：CSLU 開始（CSLU インターフェイス）

CSLU では、デバイスからの使用状況レポートの収集を手動でトリガーすることもできます。

製品インスタンスを設定して選択した後（[Add Single Product Instance] を選択し、ホスト名を入力して CSLU 開始型接続メソッドを選択）、[Actions for Selected] > [Collect Usage] を選択します。CSLU は選択した製品インスタンスに接続し、使用状況レポートを収集します。収集された使用状況レポートは、CSLU のローカルライブラリに保存されます。これらのレポートは、CSLU がシスコに接続されている場合はシスコに転送できます。または（シスコに接続されていない場合は）[Product Instances] > [Export to CSSM] の順に選択して、手動で使用状況の収集をトリガーできます。

CSLU 開始モードで作業している場合は、次の手順を実行して、製品インスタンスから RUM レポートを収集するように CSLU を設定します。

手順

ステップ 1 [Preferences] タブをクリックし、有効な [Smart Account] と [Virtual Account] を入力して、適切な CSLU 開始型収集メソッドを選択します。（[Preferences] に変更があった場合は、[Save] をクリックします）。

ステップ 2 [Inventory] タブをクリックし、1 つまたは複数の製品インスタンスを選択します。

ステップ 3 [Actions for Selected] > [Collect Usage] をクリックします。

RUM レポートは、選択した各デバイスから取得され、CSLU ローカルライブラリに保存されます。[Last Contact] 列が更新され、レポートが受信された時刻が表示されます。[Alerts] 列にはステータスが表示されます。

CSLU が現在シスコにログインしている場合、レポートはシスコの関連するスマートアカウントとバーチャルアカウントに自動的に送信され、シスコはCSLUと製品インスタンスに確認応答を送信します。確認応答は、[Product Instance] テーブルの [Alerts] 列に表示されます。

シスコに手動で使用状況レポートを転送するには、CSLU のメイン画面から [Data] > [Export to CSSM] を選択します。

ステップ 4 [Export to Cisco] モーダルから、レポートを保存するローカルディレクトリを選択できます。（<CSLU_WORKING_Directory>/data/default/rum/unsent）

この時点で、使用状況レポートがローカルディレクトリ（ライブラリ）に保存されます。使用状況レポートをシスコにアップロードするには、CSSMへの使用状況データのアップロードとACK のダウンロード （260 ページ） の手順に従ってください。

（注） Windows オペレーティングシステムでは、ファイルの名前が変更されたときに拡張子をドロップすることで、使用状況レポートファイルのプロパティの動作を変更できます。動作の変更は、ダウンロードしたファイルの名前を変更し、名前を変更したファイルが拡張子をドロップすると発生します。たとえば、`UD_xxx.tar` という名前のダウンロード済みデフォルトファイルの名前が `UD_yyy` に変更されたとします。ファイルは tar 拡張子を失い、機能しなくなります。使用状況ファイルを正常に機能させるには、使用状況レポートファイルの名前を変更した後、`UD_yyy.tar` のように、ファイル名に tar 拡張子を追加する必要があります。

CSSM へのエクスポート（CSLU インターフェイス）

このオプションは、セキュリティのためにワークステーションを隔離する場合に、手動ダウンロード手順の一部として使用できます。

手順

ステップ 1 [Preferences] タブに移動し、[Cisco Connectivity] トグルスイッチをオフにします。

フィールドが「Cisco Is Not Available」に切り替わります。

ステップ 2 CSLU のホーム画面から、[Data] > [Export to CSSM] に移動します。

ステップ 3 開いたウィンドウからファイルを選択し、[Save] をクリックします。これでファイルが保存されました。

（注） この時点で、DLC ファイル、RUM ファイル、またはその両方があります。

ステップ 4 シスコに接続できるワークステーションから、次の手順を実行します。 CSSM への使用状況データのアップロードと ACK のダウンロード （260 ページ）

ファイルがダウンロードされたら、CSLUにインポートできます。を参照してください。CSSM からのインポート（CSLU インターフェイス） （216 ページ）

CSSM からのインポート（CSLU インターフェイス）

シスコから ACK またはその他のファイル（承認コードなど）を受信すると、そのファイルをシステムにアップロードできます。この手順は、オフラインのワークステーションに使用できます。シスコからファイルを選択してアップロードするには、次の手順を実行します。

手順

ステップ 1 CSLU にアクセス可能な場所にファイルがダウンロードされていることを確認します。

ステップ 2 CSLU のホーム画面から、[Data] > [Import from CSSM] に移動します。

ステップ 3 [Import from CSSM] モーダルが開き、次のいずれかを実行できます。

- ローカルドライブにある**ファイル**をドラッグアンドドロップします。または、
- 適切な *.xml ファイルを参照し、ファイルを選択して [Open] をクリックします。

アップロードが成功すると、ファイルがサーバーに正常に送信されたことを示すメッセージが表示されます。アップロードが成功しない場合は、インポートエラーが発生します。

ステップ 4 アップロードが完了したら、ウィンドウの右上隅にある [x] をクリックして閉じます。

CSLU 開始型通信のネットワーク到達可能性の確認

このタスクでは、CSLU 開始型通信のネットワーク到達可能性を確認するために必要になる可能性のある設定を提供します。「（必須）」と付いている手順は、すべての製品インスタンスで必須です。他のすべての手順は、製品インスタンスの種類とネットワーク要件に応じて、必須の場合も任意の場合もあります。該当するコマンドを設定します。

始める前に

サポートされるトポロジ：CSLU を介して CSSM に接続（CSLU 開始型通信）。

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 モードを有効にします。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	**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	**aaa new model** 例： `Device(config)# aaa new model`	（必須）認証、許可、アカウンティング（AAA）アクセスコントロールモデルをイネーブルにします。
ステップ 4	**aaa authentication login default local** 例： `Device(config)# aaa authentication login default local`	（必須）認証時にローカルのユーザー名データベースを使用するように、AAA 認証を設定します。
ステップ 5	**aaa authorization exec default local** 例： `Device(config)# aaa authorization exec default local`	ネットワークへのユーザーアクセスを制限するパラメータを設定します。ユーザーは EXEC シェルの実行が許可されます。
ステップ 6	ip routing 例： `Device(config)# ip routing`	IP ルーティングを有効にします。
ステップ 7	{**ip** | **ipv6**}**name-server** *server-address 1 ...server-address 6*] 例： `Device(config)# ip name-server vrf Mgmt-vrf 192.168.1.100 192.168.1.200 192.168.1.300`	（任意）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ーのアドレスを指定します。 最大 6 つのネーム サーバーを指定できます。各サーバー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ーが、プライマリ サーバーです。デバイスは、プライマリサーバーへDNS クエリを最初に送信します。そのクエリが失敗した場合は、バックアップ サーバーにクエリが送信されます。
ステップ 8	**ip domain lookup source-interface** *interface-type-number* 例： `Device(config)# ip domain lookup source-interface gigabitethernet0/0`	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有効にします。この機能は、デフォルトでイネーブルにされています。 ユーザー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

	コマンドまたはアクション	目的
		て、ユーザー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。
ステップ 9	**ip domain name** *name* 例： `Device(config)# ip domain name vrf Mgmt-vrf 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。
ステップ 10	**no username** *name* 例： `Device(config)# no username admin`	（必須）指定されたユーザー名が存在する場合はクリアします。*name* には、次のステップで作成するユーザー名と同じものを入力します。これにより、次のステップで作成するユーザー名が重複していないことが保証されます。 CSLU 開始型の RUM レポート取得に REST API を使用する場合は、CSLU にログインする必要があります。ここでユーザー名が重複していると、システムにユーザー名が重複している場合にこの機能が正しく動作しないことがあります。
ステップ 11	**username** *name* **privilege** *level* **password** *password* 例： `Device(config)# username admin privilege 15 password 0 lab`	（必須）ユーザー名をベースとした認証システムを構築します。 **privilege** キーワードにより、ユーザーの権限レベルを設定します。ユーザーの権限レベルを指定する 0 ～ 15 の数字です。 password を使用すると、name 引数にアクセスできます。パスワードは 1 ～ 25 文字で、埋め込みスペースを使用でき、**username** コマンドの最後のオプションとして指定します。 これにより、CSLU が製品インスタンスのネイティブ REST を使用できるようになります。

	コマンドまたはアクション	目的
		（注） このユーザー名とパスワードをCSLUで入力します（使用状況レポートの収集：CSLU開始（CSLU インターフェイス）（214ページ）→ステップ4.f）。その後、CSLUは製品インスタンスからRUMレポートを収集できます。
ステップ12	**interface** *interface-type-number* 例： `Device (config)# interface gigabitethernet0/0`	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、VRFに関連付けられたイーサネットインターフェイス、サブインターフェイス、またはVLANを指定します。
ステップ13	**vrf forwarding** *vrf-name* 例： `Device(config-if)# vrf forwarding Mgmt-vrf`	VRFをレイヤ3インターフェイスに対応付けます。このコマンドは、インターフェイスでマルチプロトコルVRFをアクティブにします。
ステップ14	**ip address** *ip-address mask* 例： `Device(config-if)# ip address 192.168.0.1 255.255.0.0`	VRFのIPアドレスを定義します。
ステップ15	**negotiation auto** 例： `Device(config-if)# negotiation auto`	インターフェイスの速度およびデュプレックスパラメータの自動ネゴシエーション動作を有効にします。
ステップ16	**no shutdown** 例： `Device(config-if)# no shutdown`	無効にされたインターフェイスを再起動します。
ステップ17	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーション モードを終了し、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18	**ip http server** 例： `Device(config)# ip http server`	（必須）シスコのWebブラウザ ユーザー インターフェイスを含むIPまたはIPv6システムでHTTPサーバーを有効にします。HTTPサーバーは、デフォルトにより標準のポート80を使用します。

	コマンドまたはアクション	目的
ステップ 19	**ip http authentication local** 例： `ip http authentication local` `Device(config)#`	（必須）HTTP サーバーユーザーに対して特定の認証方法を指定します。 **local** キーワードは、認証および許可に、ローカルシステム設定で（username グローバル コンフィギュレーション コマンドによって）指定したログイン ユーザー名、パスワード、権限レベル アクセスの組み合わせを使用することを示します。
ステップ 20	**ip http secure-server** 例： `Device(config)# ip http server`	（必須）セキュア HTTP（HTTPS）サーバーを有効にします。HTTPS サーバーは、セキュア ソケット レイヤ（SSL）バージョン 3.0 プロトコルを使用します。
ステップ 21	**ip http max-connections** 例： `Device(config)# ip http max-connections 16`	（必須）HTTP サーバーへの同時最大接続数を設定します。1 ～ 16 の範囲の整数を入力します。デフォルトは 5 です。
ステップ 22	**ip tftp source-interface** *interface-type-number* 例： `Device(config)# ip tftp source-interface GigabitEthernet0/0`	TFTP 接続用の送信元アドレスとして、インターフェイスの IP アドレスを指定します。
ステップ 23	**ip route** *ip-address ip-mask subnet mask* 例： `Device(config)# ip route vrf mgmt-vrf 192.168.0.1 255.255.0.0 192.168.255.1`	製品インスタンスにルートとゲートウェイを設定します。スタティック ルートまたはダイナミック ルートのいずれかを設定できます。
ステップ 24	**logging host** 例： `Device(config)# logging host 172.25.33.20 vrf Mgmt-vrf`	リモート ホストへのシステム メッセージおよびデバッグ出力を記録します。
ステップ 25	**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、特権 EXEC 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 26	**show ip http server session-module** 例： `Device# show ip http server session-module`	（必須）HTTP 接続を確認します。出力で、SL_HTTP がアクティブであることを確認します。また、次のチェックも実行できます。 • CSLU がインストールされているデバイスから、製品インスタンスに ping できることを確認します。ping が成功すると、製品インスタンスが到達可能であることが確認されます • CSLU がインストールされているデバイスの Web ブラウザで、`https://<product-instance-ip>/` を確認します。これにより、CSLU から製品インスタンスへの REST API が期待どおりに動作することが保証されます。

1つ以上の製品インスタンスのSLACの要求（CSLUインターフェイス）

このタスクでは、CSLU で 1 つ以上の製品インスタンスの SLAC を手動で要求する方法を示します。

始める前に

サポートされているトポロジ：

- CSLU を介した CSSM への接続（製品インスタンス開始および CSLU 開始）
- CSLU は CSSM から切断（製品インスタンス開始および CSLU 開始）

手順

ステップ 1 [Inventory] タブに移動します。[Product Instances] テーブルから、承認コード要求の対象となる 1 つ以上の製品インスタンスを選択します。

ステップ 2 [Actions for Selected] メニューから、[Authorization Code Request] オプションを選択します。

[Authorization Request Information] のポップアップウィンドウが表示されます。

ステップ 3 [承認（Accept）] をクリックします。

アップロードする .csv ファイルを選択する別のポップアップウィンドウが開きます。

ステップ 4 ファイルをCSSMにアップロードし、承認コードを生成して、コードを含むファイルをダウンロードします。CSSM からの SLAC の生成とファイルへのダウンロード （249 ページ）を参照してください。

ステップ 5 CSLU インターフェイスに戻ります。

ステップ 6 [Data] > [Import from CSSM] を選択して、承認コードを適用します。「CSSM からのインポート（CSLU インターフェイス） （216 ページ）」を参照してください

CSLU が製品開始モードの場合：製品インスタンスが次回 CSLU に接続したときに、アップロードされたコードが製品インスタンスに適用されます。

CSLU が CSLU 開始モードの場合：CSLU が次回更新を実行するときに、アップロードされたコードが製品インスタンスに適用されます。

CSSM への接続の設定

次の手順では、CSSM へのレイヤ 3 接続を設定してネットワーク到達可能性を確認する方法を説明します。「（必須）」と付いている手順は、すべての製品インスタンスで必須です。他のすべての手順は、製品インスタンスの種類とネットワーク要件に応じて、必須の場合も任意の場合もあります。該当するコマンドを設定します。

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 モードを有効にします。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	**{ip | ipv6}name-server** *server-address 1 ...server-address 6*] 例： `Device(config)# ip name-server 209.165.201.1 209.165.200.225 209.165.201.14 209.165.200.230`	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 サーバーのアドレスを指定します。 最大 6 つのネーム サーバーを指定できます。各サーバー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ーが、プライマリ サーバーです。デバイスは、プライマリサーバーへDNS クエリを最初に送信します。そのクエリが失敗した場合は、バックアップ サーバーにクエリが送信されます。

	コマンドまたはアクション	目的
ステップ 4	**ip name-server vrf Mgmt-vrf** *server-address 1...server-address 6* 例： `Device(config)# ip name-server vrf Mgmt-vrf 209.165.201.1 209.165.200.225 209.165.201.14 209.165.200.230`	（任意）VRF インターフェイスで DNS を設定します。最大 6 つのネーム サーバーを指定できます。各サーバーアドレスはスペースで区切ります。 （注） このコマンドは、**ip name-server** コマンドの代わりです。
ステップ 5	**ip domain lookup source-interface** *interface-type interface-number* 例： `Device(config)# ip domain lookup source-interface Vlan100`	DNS ドメインルックアップ用のソースインターフェイスを設定します。
ステップ 6	**ip domain name** *domain-name* 例： `Device(config)# ip domain name example.com`	ドメイン名を設定します。
ステップ 7	**ip host tools.cisco.com** *ip-address* 例： `Device(config)# ip host tools.cisco.com 209.165.201.30`	自動 DNS マッピングが使用できない場合は、DNS ホスト名キャッシュ内のホスト名/アドレス静的マッピングを設定します。
ステップ 8	**interface** *interface-type-number* 例： `Device(config)# interface Vlan100` `Device(config-if)# ip address 192.0.2.10 255.255.255.0` `Device(config-if)# exit`	レイヤ 3 インターフェイスを設定します。インターフェイスのタイプと番号、または VLAN を入力します。
ステップ 9	**ntp server** *ip-address* [**version** *number*] [**key** *key-id*] [**prefer**] 例： `Device(config)# ntp server 198.51.100.100 version 2 prefer`	（必須）NTP サービスをアクティブにし（まだアクティブになっていない場合）、システムがシステム ソフトウェア クロックを指定された NTP サーバーと同期できるようにします。これにより、デバイスの時刻が CSSM と同期されます。 このコマンドを複数回使用する必要があるために優先サーバーを設定する場合は、**prefer** キーワードを使用します。このキーワードを使用すると、サーバー間の切り換え回数が減少します。

	コマンドまたはアクション	目的
ステップ 10	**switchport access vlan** *vlan_id* 例： `Device(config)# interface GigabitEthernet1/0/1` `Device(config-if)# switchport access vlan 100` `Device(config-if)# switchport mode access` `Device(config-if)# exit` `OR` `Device(config)#`	このアクセスポートがトラフィックを伝送する VLAN を有効にし、非トランキングで非タグ付きのシングル VLAN イーサネットインターフェイスとしてインターフェイスを設定します。 （注） このステップは、スイッチポート アクセス モードが必要な場合にのみ設定します。**switchport access vlan** コマンドは、たとえば Catalyst スイッチング製品インスタンスに適用できます。ルーティング製品インスタンスの場合は、代わりに **ip address** *ip-address mask* コマンドを設定できます。
ステップ 11	**ip route** *ip-address ip-mask subnet mask* 例： `Device(config)# ip route 192.0.2.0 255.255.255.255 192.0.2.1`	デバイスにルートを設定します。スタティックルートまたはダイナミックルートのいずれかを設定できます。
ステップ 12	**ip http client source-interface** *interface-type-number* 例： `Device(config)# ip http client source-interface Vlan100`	（必須）HTTP クライアントのソースインターフェイスを設定します。インターフェイスのタイプと番号、または VLAN を入力します。
ステップ 13	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 14	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	コンフィギュレーションファイルに設定を保存します。

HTTPS プロキシを介したスマート転送の設定

スマート転送モードを使用している場合にプロキシサーバーを使用して CSSM と通信するには、次の手順を実行します。

（注） 認証された HTTPS プロキシ設定はサポートされていません。

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 モードを有効にします。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	**license smart transport smart** 例： `Device(config)# license smart transport smart`	スマート転送モードを有効にします。
ステップ 4	**license smart url default** 例： `Device(config)# license smart transport default`	スマート URL を自動的に設定します（https://smartreceiver.cisco.com/licservice/license）。このオプションを想定どおりに動作させるには、前の手順の転送モードを smart に設定する必要があります。
ステップ 5	**license smart proxy {address** *address_hostname* **| port** *port_num*} 例： `Device(config)# license smart proxy 198.51.100.10 port 3128`	スマート転送モードのプロキシを設定します。プロキシが設定されている場合、ライセンスメッセージは最終宛先 URL（CSSM）に加えてプロキシにも送信されます。プロキシはメッセージを CSSM に送信します。アドレスとポート情報を入力します。 • **address** *address_hostname*：プロキシ アドレスを指定します。プロキシ サーバーの IP アドレスまたはホスト名を入力します。 • **port** *port_num*：プロキシポートを指定します。プロキシ ポートポート番号を入力します。 Cisco IOS XE Bengaluru 17.6.1 以降、プロキシサーバーの受け入れ基準が変更されたことに注意してください。プロキシ

	コマンドまたはアクション	目的
		サーバーの応答のステータスコードのみがシステムによって検証され、理由フレーズは検証されません。RFC形式は、`status-line = HTTP-version SP status-code SP reason-phrase CRLF` です。ステータス行の詳細については、RFC 7230 のセクション 3.1.2 を参照してください。
ステップ 6	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	コンフィギュレーション ファイルに設定を保存します。

ダイレクトクラウドアクセス用の Call Home サービスの設定

Call Home サービスは、CSSM に対してクリティカルなシステムイベントを電子メールおよび Web 上で通知します。転送モードを設定するには、Call Home サービスを有効にし、宛先プロファイルを設定して（宛先プロファイルには、アラート通知に必要な配信情報が含まれます。少なくとも 1 つの宛先プロファイルが必要です）、次の手順を実行します。

（注） 「（任意）」と特に明記されていない限り、すべての手順を実行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	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。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	**license smart transport callhome** 例：	転送モードとして Call Home を有効にします。

	コマンドまたはアクション	目的
	`Device(config)# license smart transport callhome`	
ステップ 4	**license smart url** *url* 例： `Device(config)# license smart url https://tools.cisco.com/its/service/oddce/services/DDCEService`	**callhome** 転送モードの場合は、例に示すように CSSM URL を設定します。
ステップ 5	**service call-home** 例： `Device(config)# service call-home`	Call Home 機能をイネーブルにします。
ステップ 6	**call-home** 例： `Device(config)# call-home`	Call Home コンフィギュレーション モードを開始します。
ステップ 7	**contact-email-address** *email-address* 例： `Device(config-call-home)# contact-email-addr username@example.com`	お客様の電子メールアドレスを割り当て、Smart Call Home サービスのフルレポート機能を有効にし、フルインベントリメッセージを Call Home TAC プロファイルから Smart Call Home サーバーに送信してフル登録プロセスを開始します。電子メールアドレスフォーマットには、スペースなしで最大 200 文字まで入力できます。
ステップ 8	**profile** *name* 例： `Device(config-call-home)# profile CiscoTAC-1` `Device(config-call-home-profile)#`	指定された宛先プロファイルに対する Call Home 宛先プロファイル設定サブモードに入ります。 デフォルトは次のとおりです。 • CiscoTAC-1 プロファイルは非アクティブです。このプロファイルを Call Home サービスで使用するには、プロファイルを有効にする必要があります。 • CiscoTAC-1 プロファイルは、プロファイルに登録されているすべてのイベントタイプが記載された完全なレポートを送信します。または、`Device(cfg-call-home-profile)# anonymous-reporting-only` anonymous-reporting-only を追加で設定します。これが設定されてい

	コマンドまたはアクション	目的
		る場合は、クラッシュ、インベントリ、およびテストメッセージのみが送信されます。 プロファイルのステータスを確認するには、**show call-home profile all** コマンドを使用します。
ステップ 9	**active** 例： `Device(config-call-home-profile)# active`	宛先プロファイルをイネーブルにします。
ステップ 10	**destination transport-method http**{**email**|**http**} 例： `Device(config-call-home-profile)# destination transport-method http` AND `Device(config-call-home-profile)# no destination transport-method email`	メッセージの転送形式をイネーブルにします。この例では、HTTP経由でCall Homeサービスが有効になり、電子メールによる転送が無効になります。 このコマンドのno形式を使用すると、メソッドが無効になります。
ステップ 11	**destination address** { **email** *email_address*|**http** *url*} 例： `Device(config-call-home-profile)# destination address http https://tools.cisco.com/its/service/oddce/services/DDCEService` AND `Device(config-call-home-profile)# no destination address http https://tools.cisco.com/its/service/oddce/services/DDCEService`	Call Home メッセージを送信する宛先 E メール アドレスまたは URL を設定します。宛先URL を入力する場合は、サーバーがセキュアサーバーであるかどうかに応じて http://（デフォルト）または https:// を指定します。 ここに示す例では、http:// の形式で宛先URL が設定されています。コマンドの **no** 形式ではhttps:// に設定されます。
ステップ 12	**exit** 例： `Device(config-call-home-profile)# exit`	Call Home 宛先プロファイル コンフィギュレーションモードを終了して、Call Home コンフィギュレーション モードに戻ります。
ステップ 13	**exit** 例： `Device(config-call-home)# end`	Call Home コンフィギュレーションモードを終了して、特権 EXEC モードに戻ります。
ステップ 14	**copy running-config startup-config** 例：	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
	`Device# copy running-config startup-config`	
ステップ 15	**show call-home profile** {*name* |**all**}	指定されたプロファイル、または設定済みのすべてのプロファイルに関する宛先プロファイル設定を表示します。

HTTPS プロキシサーバーを介したダイレクトクラウドアクセス用の Call Home サービスの設定

Call Home サービスは、HTTPS プロキシサーバーを介して設定できます。この設定では、CSSM への接続にユーザー認証は必要ありません。

（注） 認証された HTTPS プロキシ設定はサポートされていません。

HTTPS プロキシを介して Call Home サービスを設定して有効にするには、次の手順を実行します。

（注） 「（任意）」と特に明記されていない限り、すべての手順を実行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	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。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	**license smart transport callhome** 例： `Device(config)# license smart transport callhome`	転送モードとして Call Home を有効にします。
ステップ 4	**service call-home** 例： `Device(config)# service call-home`	Call Home 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
ステップ 5	**call-home** 例： `Device(config)# call-home`	Call Home コンフィギュレーション モードを開始します。
ステップ 6	**http-proxy** *proxy-address* **proxy-port** *port-number* 例： `Device(config-call-home)# http-proxy 198.51.100.10 port 5000`	Call Home サービスへのプロキシサーバー情報を設定します。
ステップ 7	**exit** 例： `Device(config-call-home)# exit`	Call Home コンフィギュレーション モードを終了し、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。 Cisco IOS XE Bengaluru 17.6.1 以降、プロキシサーバーの受け入れ基準が変更されたことに注意してください。プロキシサーバーの応答のステータスコードのみがシステムによって検証され、理由フレーズは検証されません。RFC形式は、`status-line = HTTP-version SP status-code SP reason-phrase CRLF` です。ステータス行の詳細については、RFC 7230 のセクション 3.1.2 を参照してください。
ステップ 8	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードを開始します。
ステップ 9	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	コンフィギュレーション ファイルに設定を保存します。

スマートアカウントとバーチャルアカウントの割り当て（SSMオンプレミス UI）

この手順を使用して、1 つ以上の製品インスタンスを対応するスマートアカウントおよびバーチャルアカウント情報とともに SSM オンプレミスのデータベースにインポートできます。これにより、SSM オンプレミスは、ローカルバーチャルアカウント（デフォルトのローカルバーチャル アカウント以外）の一部である製品インスタンスを CSSM の正しいライセンスプールにマッピングできます。

始める前に

サポートされているトポロジ：SSM オンプレミス展開（製品スタンス開始型通信）。

手順

ステップ 1 SSM オンプレミスにログインし、[Smart Licensing] ワークスペースを選択します。

ステップ 2 [Inventory] > [SL Using Policy] > [Export/Import All] > [Import Product Instances List] に移動します。

[Upload Product Instances] ウィンドウが表示されます。

ステップ 3 [Download] をクリックして .csv テンプレートファイルをダウンロードし、テンプレート内のすべての製品インスタンスに必要な情報を入力します。

ステップ 4 テンプレートに入力したら、[Inventory] > [SL Using Policy] > [Export/Import All] > [Import Product Instances List] をクリックします。

[Upload Product Instances] ウィンドウが表示されます。

ステップ 5 [Browse] をクリックし、入力した .csv テンプレートをアップロードします。

アップロードしたすべての製品インスタンスのスマートアカウント情報とバーチャルアカウント情報が SSM オンプレミスで使用できるようになりました。

デバイスの検証（SSM オンプレミス UI）

デバイス検証が有効になっている場合、不明な製品インスタンス（SSM オンプレミスデータベース内にない）からの RUM レポートは拒否されます。

デフォルトでは、デバイスは検証されません。この機能を有効にするには、次の手順を実行します。

始める前に

サポートされているトポロジ：SSM オンプレミス展開（製品スタンス開始型通信）。

手順

ステップ 1 [On-Prem License Workspace] ウィンドウで、[Admin Workspace] をクリックし、プロンプトが表示されたらログインします。

[On-Prem Admin Workspace] ウィンドウが表示されます。

ステップ 2 [Settings] ウィジェットをクリックします。

[Settings] ウィンドウが表示されます。

ステップ 3 [CSLU] タブに移動し、[Validate Device] トグルスイッチをオンにします。

不明な製品インスタンスからの RUM レポートが拒否されるようになりました。必要な製品インスタンスを SSM オンプレミスデータベースにまだ追加していない場合は、RUM レポートを送信する前に追加する必要があります。「スマートアカウントとバーチャルアカウントの割り当て（SSM オンプレミス UI）（230 ページ）」を参照してください。

製品インスタンス開始型通信のネットワーク到達可能性の確認

このタスクでは、製品インスタンス開始型通信のネットワーク到達可能性を確認するために必要になる可能性のある設定を提供します。「（必須）」と付いている手順は、すべての製品インスタンスで必須です。他のすべての手順は、製品インスタンスの種類とネットワーク要件に応じて、必須の場合も任意の場合もあります。該当するコマンドを設定します。

（注） 手順 13、14、および 15 では、必ず次のように設定してください。これらのコマンドは、正しいトラストポイントが使用され、ネットワーク到達可能性に必要な証明書が受け入れられるように設定する必要があります。

始める前に

サポートされているトポロジ：SSM オンプレミス展開（製品スタンス開始型通信）。

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 モードを有効にします。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	**interface** *interface-type-number* 例： `Device (config)# interface gigabitethernet0/0`	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、VRF に関連付けられたイーサネットインターフェイス、サブインターフェイス、または VLAN を指定します。
ステップ 4	**vrf forwarding** *vrf-name* 例：	VRF をレイヤ 3 インターフェイスに対応付けます。このコマンドは、イン

	コマンドまたはアクション	目的
	`Device(config-if)# vrf forwarding Mgmt-vrf`	ターフェイスでマルチプロトコルVRFをアクティブにします。
ステップ5	**ip address** *ip-address mask* 例： `Device(config-if)# ip address 192.168.0.1 255.255.0.0`	VRF の IP アドレスを定義します。
ステップ6	**negotiation auto** 例： `Device(config-if)# negotiation auto`	インターフェイスの速度およびデュプレックスパラメータの自動ネゴシエーション動作を有効にします。
ステップ7	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーションモードを終了し、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8	**ip http client source-interface** *interface-type-number* 例： `Device(config)# ip http client source-interface gigabitethernet0/0`	HTTP クライアントのソース インターフェイスを設定します。
ステップ9	**ip route** *ip-address ip-mask subnet mask* 例： `Device(config)# ip route vrf mgmt-vrf 192.168.0.1 255.255.0.0 192.168.255.1`	（必須）製品インスタンスにルートとゲートウェイを設定します。スタティックルートまたはダイナミックルートのいずれかを設定できます。
ステップ10	{**ip** | **ipv6**}**name-server** *server-address 1 ...server-address 6*] 例： `Device(config)# Device(config)# ip name-server vrf mgmt-vrf 198.51.100.1`	VRFインターフェイスでドメインネームシステム（DNS）を設定します。
ステップ11	**ip domain lookup source-interface** *interface-type-number* 例： `Device(config)# ip domain lookup source-interface gigabitethernet0/0`	DNSドメインルックアップ用のソースインターフェイスを設定します。
ステップ12	**ip domain name** *domain-name* 例： `Device(config)# ip domain name example.com`	ドメインのDNSディスカバリを設定します。この例では、ネームサーバーがエントリ `cslu-local.example.com` を作成します。

	コマンドまたはアクション	目的
ステップ 13	**crypto pki trustpoint SLA-TrustPoint** 例： `Device(config)# crypto pki trustpoint SLA-TrustPoint` `Device(ca-trustpoint)#`	（必須）製品インスタンスがトランスポイント「SLA-TrustPoint」を使用する必要があることを宣言し、CA トランスポイント コンフィギュレーション モードを開始します。このコマンドを使用してトラストポイントを宣言するまで、製品インスタンスはトラストポイントを認識しません。
ステップ 14	enrollment terminal 例： `Device(ca-trustpoint)# enrollment terminal`	（必須）証明書登録方式を指定します。
ステップ 15	**revocation-check none** 例： `Device(ca-trustpoint)# revocation-check none`	（必須）ピアの証明書が失効していないことを確認するために使用する方法を指定します。SSM オンプレミス展開トポロジの場合は、**none** キーワードを入力します。つまり、失効チェックは実行されず、証明書は常に受け入れられます。
ステップ 16	**exit** 例： `Device(ca-trustpoint)# exit` `Device(config)# exit`	CA トランスポイント コンフィギュレーション モードを終了し、次にグローバル コンフィギュレーション モードを終了してから、特権 EXEC モードに戻ります。
ステップ 17	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	コンフィギュレーション ファイルに設定を保存します。

トランスポート URL の取得（SSM オンプレミス UI）

製品インスタンス開始型通信を SSM オンプレミス展開で展開するときに、製品インスタンスでトランスポート URL を設定する必要があります。このタスクでは、テナント ID を含む完全な URL を SSM オンプレミスから簡単にコピーする方法を示します。

始める前に

サポートされているトポロジ：SSM オンプレミス展開（製品スタンス開始型通信）。

手順

ステップ 1 SSM オンプレミスにログインし、[Smart Licensing] ワークスペースを選択します。

ステップ 2 [Inventory] タブに移動し、ローカルバーチャルアカウントのドロップダウンリスト（右上隅）から、デフォルトのローカルバーチャルアカウントを選択します。この場合、[Inventory] タブの下の領域に [Local Virtual Account: Default] が表示されます。

ステップ 3 [General] タブに移動します。

[Product Instance Registration Tokens] 領域が表示されます。

ステップ 4 [Product Instance Registration Tokens] 領域で、[CSLU Transport URL] をクリックします。

[Product Registration URL] ポップアップウィンドウが表示されます。

ステップ 5 URL 全体をコピーし、アクセス可能な場所に保存します。

製品インスタンスでトランスポートタイプと URL を設定するときに、この URL が必要になります。

ステップ 6 トランスポートタイプと URL を設定します。転送タイプ、URL、およびレポート間隔の設定（262 ページ）を参照してください。

使用状況データのエクスポートとインポート（SSMオンプレミスUI）

SSM オンプレミスが CSSM から切断されている場合は、この手順を使用して SSMオンプレミスと CSSM との間で使用状況の同期を実行できます。

始める前に

サポートされているトポロジ：

- SSM オンプレミス展開（SSM オンプレミス開始型通信）
- SSM オンプレミス展開（製品インスタンス開始型通信）。

レポートデータは、SSMオンプレミスで使用できる必要があります。必要なレポートデータを製品インスタンスからSSMオンプレミスにプッシュする（製品インスタンス開始型通信）か、または必要なレポートデータを製品インスタンスから取得する（SSM オンプレミス開始型通信）必要があります。

手順

ステップ 1 SSM オンプレミスにログインし、[Smart Licensing] を選択します。

ステップ 2 [Inventory] > [SL Using Policy] タブに移動します。

ステップ 3 [SL Using Policy] タブ領域で、[Export/Import All ...] > [Export Usage to Cisco] をクリックします。

これにより、SSM オンプレミスサーバーで使用可能なすべての使用状況レポートを含む .tar ファイルが 1 つ生成されます。

ステップ 4 CSSM でCSSM への使用状況データのアップロードと ACK のダウンロード（260 ページ）のタスクを実行します。

このタスクの最後に、SSM オンプレミスにインポートする ACK ファイルを取得します。

ステップ 5 再度、[Inventory] > [SL Using Policy] タブに移動します。

ステップ 6 [SL Using Policy] タブ領域で、[Export/Import All ...] > [Import From Cisco] をクリックします。.tar ACK ファイルをアップロードします。

ACK インポートを確認するには、[SL Using Policy] タブ領域で、対応する製品インスタンスの [Alerts] 列を確認します。「Acknowledgmentreceived from CSSM」というメッセージが表示されます。

1 つ以上の製品インスタンスの追加（SSM オンプレミス UI）

次の手順を使用して、1 つの製品インスタンスを追加したり、複数の製品インスタンスをインポートして追加したりできます。これにより、SSM オンプレミスは製品インスタンスから情報を取得できるようになります。

始める前に

サポートされているトポロジ：SSM オンプレミス展開（SSM オンプレミス開始型通信）。

手順

ステップ 1 SSM オンプレミス UI にログインし、[Smart Licensing] をクリックします。

ステップ 2 [Inventory] タブに移動します。右上隅にあるドロップダウンリストからローカル バーチャル アカウントを選択します。

ステップ 3 [SL Using Policy] に移動します。

ステップ 4 単一の製品インスタンスを追加するか、または複数の製品インスタンスをインポートします（いずれかを選択します）。

- **単一の製品インスタンスを追加**するには、次の手順を実行します。
  1. [SL Using Policy] タブ領域で、[Add Single Product] をクリックします。
  2. [Host] フィールドにホストの IP アドレスを入力します（製品インスタンス）。
  3. [Connect Method] ドロップダウンリストから、適切な SSM オンプレミス開始型の接続方式を選択します。

     SSM オンプレミス開始型通信に使用できる接続方法は、NETCONF、RESTCONF、および REST API です。

4. 右側のパネルで、[Product Instance Login Credentials] をクリックします。

   [Product Instance Login Credentials] ウィンドウが表示されます。

   （注） 製品インスタンスに SLAC が必要な場合は、ログインクレデンシャルのみが必要です。

5. [User ID] と [Password] に入力し、[Save] をクリックします。

   これは、ネットワーク到達可能性を確立するために必要なコマンドの一部として設定したものと同じユーザー ID とパスワードです（SSM オンプレミス開始型通信のネットワーク到達可能性の確保 （237 ページ））。

   検証が完了すると、製品インスタンスが [SL Using Policy] タブ領域のリストに表示されます。

- **複数の製品インスタンスをインポート**するには、次の手順を実行します。

1. [SL Using Policy] タブで、[Export/Import All ...] > [Import Product Instances List] をクリックします。

   [Upload Product Instances] ウィンドウが表示されます。

2. [Download] をクリックし、事前に定義した .csv テンプレートをダウンロードします。

3. .csv テンプレートのすべての製品インスタンスに必要な情報を入力します。

   テンプレートで、すべての製品インスタンスの [Host]、[Connect Method]、および [Login Credentials] を必ず指定してください。

   SSM オンプレミス開始型通信に使用できる接続方法は、NETCONF、RESTCONF、および REST API です。

   ログインクレデンシャルは、ネットワーク到達可能性を確立するために必要なコマンドの一部として設定したユーザー ID とパスワードを参照します（SSM オンプレミス開始型通信のネットワーク到達可能性の確保 （237 ページ））。

4. 再度、[Inventory] > [SL Using Policy] タブに移動します。[Export/Import All....] > [Import Product Instances List] をクリックします。

   [Upload Product Instances] ウィンドウが表示されます。

5. 次に、入力した .csv テンプレートをアップロードします。

   検証されると、製品インスタンスが [SL Using Policy] タブのリストに表示されます。

SSM オンプレミス開始型通信のネットワーク到達可能性の確保

このタスクでは、SSM オンプレミス開始型通信のネットワーク到達可能性を確保するために必要になる可能性のある設定を実行します。「（必須）」と付いている手順は、すべての製品イ

ンスタンスで必須です。他のすべての手順は、製品インスタンスの種類とネットワーク要件に応じて、必須の場合も任意の場合もあります。該当するコマンドを設定します。

（注） 手順 25、26、および 27 では、必ず次のように設定してください。これらのコマンドは、正しいトラストポイントが使用され、ネットワーク到達可能性に必要な証明書が受け入れられるように設定する必要があります。

始める前に

サポートされているトポロジ：SSM オンプレミス展開（SSM オンプレミス開始型通信）。

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 モードを有効にします。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	**aaa new model** 例： `Device(config)# aaa new model`	（必須）認証、許可、アカウンティング（AAA）アクセスコントロールモデルをイネーブルにします。
ステップ 4	**aaa authentication login default local** 例： `Device(config)# aaa authentication login default local`	（必須）認証時にローカルのユーザー名データベースを使用するように、AAA 認証を設定します。
ステップ 5	**aaa authorization exec default local** 例： `Device(config)# aaa authorization exec default local`	ネットワークへのユーザーアクセスを制限するパラメータを設定します。ユーザーは EXEC シェルの実行が許可されます。
ステップ 6	ip routing 例： `Device(config)# ip routing`	IP ルーティングを有効にします。
ステップ 7	{**ip** | **ipv6**}**name-server** *server-address 1 ...server-address 6*] 例：	（任意）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ーのアドレスを指定します。

	コマンドまたはアクション	目的
	`Device(config)# ip name-server vrf Mgmt-vrf 192.168.1.100 192.168.1.200 192.168.1.300`	最大 6 つのネーム サーバーを指定できます。各サーバー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ーが、プライマリ サーバーです。デバイスは、プライマリサーバーへDNSクエリを最初に送信します。そのクエリが失敗した場合は、バックアップ サーバーにクエリが送信されます。
ステップ 8	**ip domain lookup source-interface** *interface-type-number* 例： `Device(config)# ip domain lookup source-interface gigabitethernet0/0`	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有効にします。この機能は、デフォルトでイネーブルにされています。 ユーザー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、ユーザー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。
ステップ 9	**ip domain name** *name* 例： `Device(config)# ip domain name vrf Mgmt-vrf 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。
ステップ 10	**no username** *name* 例： `Device(config)# no username admin`	（必須）指定されたユーザー名が存在する場合はクリアします。*name* には、次のステップで作成するユーザー名と同じものを入力します。これにより、次のステップで作成するユーザー名が重複していないことが保証されます。 SSM オンプレミス開始型の RUM レポートを取得に REST API を使用する場合は、SSMオンプレミスにログインする必要があります。ユーザー名が重複していると、システムにそのユーザー名がある場合はこの機能が正しく動作しない場合があります。

	コマンドまたはアクション	目的
ステップ 11	**username** *name* **privilege** *level* **password** *password* 例： `Device(config)# username admin privilege 15 password 0 lab`	（必須）ユーザー名をベースとした認証システムを構築します。 **privilege** キーワードにより、ユーザーの権限レベルを設定します。ユーザーの権限レベルを指定する 0 ～ 15 の数字です。 password を使用すると、name 引数にアクセスできます。パスワードは 1 ～ 25 文字で、埋め込みスペースを使用でき、**username** コマンドの最後のオプションとして指定します。 これにより、SSM オンプレミスが製品インスタンスのネイティブ REST を使用できるようになります。 （注） このユーザー名とパスワードを SSM オンプレミスに入力します（1 つ以上の製品インスタンスの追加（SSM オンプレミス UI）（236 ページ））。これにより、SSM オンプレミスは製品インスタンスから RUM レポートを収集できるようになります。
ステップ 12	**interface** *interface-type-number* 例： `Device (config)# interface gigabitethernet0/0`	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、VRF に関連付けられたイーサネットインターフェイス、サブインターフェイス、または VLAN を指定します。
ステップ 13	**vrf forwarding** *vrf-name* 例： `Device(config-if)# vrf forwarding Mgmt-vrf`	VRF をレイヤ 3 インターフェイスに対応付けます。このコマンドは、インターフェイスでマルチプロトコル VRF をアクティブにします。
ステップ 14	**ip address** *ip-address mask* 例： `Device(config-if)# ip address 192.168.0.1 255.255.0.0`	VRF の IP アドレスを定義します。

	コマンドまたはアクション	目的
ステップ 15	**negotiation auto** 例： `Device(config-if)# negotiation auto`	インターフェイスの速度およびデュプレックスパラメータの自動ネゴシエーション動作を有効にします。
ステップ 16	**no shutdown** 例： `Device(config-if)# no shutdown`	無効にされたインターフェイスを再起動します。
ステップ 17	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーションモードを終了し、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 18	**ip http server** 例： `Device(config)# ip http server`	（必須）シスコの Web ブラウザ ユーザー インターフェイスを含む IP または IPv6 システムで HTTP サーバーを有効にします。HTTP サーバーは、デフォルトにより標準のポート 80 を使用します。
ステップ 19	**ip http authentication local** 例： `ip http authentication local` `Device(config)#`	（必須）HTTP サーバーユーザーに対して特定の認証方法を指定します。 **local** キーワードは、認証および許可に、ローカルシステム設定で（username グローバル コンフィギュレーション コマンドによって）指定したログイン ユーザー名、パスワード、権限レベル アクセスの組み合わせを使用することを示します。
ステップ 20	**ip http secure-server** 例： `Device(config)# ip http server`	（必須）セキュア HTTP（HTTPS）サーバーを有効にします。HTTPS サーバーは、セキュア ソケット レイヤ（SSL）バージョン 3.0 プロトコルを使用します。
ステップ 21	**ip http max-connections** 例： `Device(config)# ip http max-connections 16`	（必須）HTTP サーバーへの同時最大接続数を設定します。1 ～ 16 の範囲の整数を入力します。デフォルトは 5 です。
ステップ 22	**ip tftp source-interface** *interface-type-number* 例：	TFTP 接続用の送信元アドレスとして、インターフェイスの IP アドレスを指定します。

	コマンドまたはアクション	目的
	`Device(config)# ip tftp source-interface GigabitEthernet0/0`	
ステップ 23	**ip route** *ip-address ip-mask subnet mask* 例： `Device(config)# ip route vrf mgmt-vrf 192.168.0.1 255.255.0.0 192.168.255.1`	製品インスタンスにルートとゲートウェイを設定します。スタティックルートまたはダイナミックルートのいずれかを設定できます。
ステップ 24	**logging host** 例： `Device(config)# logging host 172.25.33.20 vrf Mgmt-vrf`	リモートホストへのシステムメッセージおよびデバッグ出力を記録します。
ステップ 25	**crypto pki trustpoint SLA-TrustPoint** 例： `Device(config)# crypto pki trustpoint SLA-TrustPoint` `Device(ca-trustpoint)#`	（必須）製品インスタンスがトランスポイント「SLA-TrustPoint」を使用する必要があることを宣言し、CA トランスポイント コンフィギュレーション モードを開始します。このコマンドを使用してトラストポイントを宣言するまで、製品インスタンスはトラストポイントを認識しません。
ステップ 26	enrollment terminal 例： `Device(ca-trustpoint)# enrollment terminal`	（必須）証明書登録方式を指定します。
ステップ 27	**revocation-check none** 例： `Device(ca-trustpoint)# revocation-check none`	（必須）ピアの証明書が失効していないことを確認するために使用する方法を指定します。SSMオンプレミス展開トポロジの場合は、**none** キーワードを入力します。つまり、失効チェックは実行されず、証明書は常に受け入れられます。
ステップ 28	**end** 例： `Device(ca-trustpoint)# exit` `Device(config)# end`	CA トランスポイントコンフィギュレーションモードを終了し、次にグローバルコンフィギュレーションモードを終了してから、特権 EXEC モードに戻ります。
ステップ 29	**show ip http server session-module** 例：	（必須）HTTP 接続を確認します。出力で、SL_HTTP がアクティブであるこ

	コマンドまたはアクション	目的
	`Device# show ip http server session-module`	とを確認します。また、次のチェックも実行できます。 • SSMオンプレミスがインストールされているデバイスから、製品インスタンスにpingできることを確認します。pingが成功すると、製品インスタンスが到達可能であることが確認されます • SSMオンプレミスがインストールされているデバイスのWebブラウザで、`https://<product-instance-ip>/` を確認します。これにより、SSMオンプレミスから製品インスタンスへのREST APIが期待どおりに動作することが保証されます。
ステップ 30	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	コンフィギュレーションファイルに設定を保存します。

承認コード要求の送信（SSM オンプレミス UI）

SSMオンプレミス展開のトポロジを使用すると、製品インスタンスが要求する前に、輸出規制ライセンスと適用済みライセンスに必要な承認コードを CSSM で生成して、SSM オンプレミスにインポートする必要があります。この手順には、SSMオンプレミスで実行する必要がある手順（要求を送信して、その後にSLACをインポートする）を説明し、CSSM で実行する必要がある手順（SLAC を生成してダウンロードする）と製品インスタンスで実行する必要がある手順（最終的に SLAC を要求してインストールする）を示します。

始める前に

サポートされているトポロジ：

- SSM オンプレミス展開（SSM オンプレミス開始型通信）
- SSM オンプレミス展開（製品インスタンス開始型通信）。

CSSMのスマートアカウントとバーチャルアカウントに、必要な輸出規制または適用済みライセンスのバランスが十分にプラスであることを確認します。

手順

ステップ 1 SSM オンプレミスにログインし、[Smart Licensing] を選択します。

ステップ 2 [Inventory] > [SL Using Policy] に移動します。SLAC を要求するすべての製品インスタンスを選択します。

ステップ 3 [Actions for Selected…] > [Authorization Code Request] をクリックします。

[Authorization Request Information] ポップアップウィンドウが表示されます。

ステップ 4 [Accept] をクリックし、プロンプトが表示されたら .csv ファイルを保存します。

generated.csv ファイルには、選択した製品インスタンスのリストが、CSSM で SLAC を生成するために必要な形式で含まれています。CSSM Web UI で作業しているときにアクセス可能な場所にこのファイルを保存します（次の手順）。

ステップ 5 CSSM でCSSM からの SLAC の生成とファイルへのダウンロード （249 ページ）のタスクを実行します。

上記の手順を使用して、単一の製品インスタンスに対しても、複数の製品インスタンスに対しても SLAC を生成できます。SSM オンプレミス展開トポロジの場合は、複数の製品インスタンスに SLAC を生成する手順に従います。

ステップ 6 再度、[Inventory] > [SL Using Policy] に移動します。

ステップ 7 [Export/Import All…] をクリックし、[Import From Cisco] をクリックします。

上記の手順 4 の最後にダウンロードした .csv ファイルをインポートします。

インポートを確認するには、[Inventory] > [SL Using Policy] の下にある [Alerts] 列を参照します。「Authorization message received from CSSM」というメッセージが表示されます。

ステップ 8 製品インスタンスまたは SSM オンプレミスが通信を開始するかどうかに応じて、最後の手順を実行します。

- 製品インスタンス開始型通信の場合、SSM オンプレミスから SLAC を要求してインストールするように製品インスタンスを設定します。次を参照してください。 SLAC の手動要求と自動インストール （244 ページ）
- SSM オンプレミス開始型通信の場合、SSM オンプレミスが次に更新を実行するときに、アップロードされたコードが製品インスタンスに適用されます。

SLAC の手動要求と自動インストール

CSSM、CSLU、または SSM オンプレミスに SLAC を要求し、製品インスタンスに自動的にインストールするには、製品インスタンスで次の手順を実行します。

始める前に

サポートされるトポロジ：

- CSLU を介した CSSM への接続（製品インスタンス開始型通信および CSLU 開始型通信）
- CSSM に直接接続
- CSLU は CSSM から切断（製品インスタンス開始型通信および CSLU 開始型通信）
- SSM オンプレミス展開（製品インスタンス開始型通信）

続行する前に、次の点も確認してください。

- SLAC を要求している製品インスタンスが CSSM、CSLU、または SSM オンプレミスに接続されています。
- トランスポートタイプと URL がそれに応じて設定されます。特権 EXEC モードで **show license all** コマンドを使用します。出力で、`Transport:` フィールドを確認します。
- CSSMに直接接続している場合は、トークンを生成することで信頼コードをインストールしています。**show license all** コマンドは特権 EXEC モードで入力します。出力で、`Trust Code Installed:` フィールドを確認します。
- SSM オンプレミス展開の場合、製品インスタンスは SLAC の SSM オンプレミスを要求するため、このタスクを開始する前に、必要な数の SLAC ファイルが SSM オンプレミスサーバーで使用可能な状態に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	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。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	**license smart authorization request** {**add** | **replace**} *feature_name* {**all** | **local**} 例： `Device# license smart authorization request add hseck9 local`	• **license smart authorization request**：このオプションは、CSSM、CSLU または SSM オンプレミスから SLAC を要求します。SLAC が返され、製品インスタンスに自動的にインストールされます。 既存の SLAC に追加するのか置換するのかを指定します。 • **add**：要求されたライセンスを既存の SLAC に追加します。新しい承認コードには、既存の SLAC のすべてのライセンスと

	コマンドまたはアクション	目的
		要求されたライセンスが含まれます。 • **replace**：既存の SLAC を置き換えます。新しい SLAC には、要求されたライセンスのみが含まれます。既存の SLAC のすべてのライセンスが返却されます。このキーワードを入力すると、製品インスタンスはこれらの既存のライセンスが使用中かどうかを確認します。使用中の場合は、対応する機能を最初に無効にするようにエラーメッセージが表示されます。 *feature_name*：SLAC の追加または置換を要求するライセンスの名前を入力します。「hseck9」と入力して、HSECK9 ライセンスの SLAC を要求してインストールします。 次のいずれかのオプションを入力して、デバイスを指定します。 • **all**：高可用性設定およびスタック設定のすべてのデバイスの承認コードを取得します。

	コマンドまたはアクション	目的
		（注） SLAC がすでにインストールされている既存のスタックに（SLAC がインストールされていない）デバイスを追加した場合は、replace および all オプションを使用します。これにより、スタック内のすべてのデバイスの SLAC が要求されます。特定のメンバーの SLAC を要求することはできません。選択肢はアクティブまたはスタック全体のみです。 • **local**：高可用性設定およびスタック設定のアクティブなデバイスの承認コードを取得します。これがデフォルトのオプションです。
ステップ 3	（任意） **license smart sync{all | local}** 例： `Device# license smart sync local`	CSSM、CSLU または SSM オンプレミスと製品インスタンスの同期がトリガーされ、保留中のデータが送受信されます。 この手順はオプションであり、製品インスタンスが CSSM、CSLU または SSM オンプレミスに接続され、製品インスタンスが通信を開始するシナリオにのみ適用されます。対応するトポロジは、CSSM に直接接続、CSLU を介して CSSM に接続（製品インスタンス開始）、および SSM オンプレミス展開（製品インスタンス開始型通信）です。 このコマンドは、手動で同期をトリガーし、SLAC インストールプロセスを完了します。それ以外の場合、製品インスタンスが次回 CSLU または SSM オンプレ

	コマンドまたはアクション	目的
		ミスに接続するときに、SLAC が製品インスタンスに適用されます。
ステップ 4	該当するトポロジの残りの手順を実行します。	• CSLU を介して CSSM に接続（CSLU 開始型通信）については、CSLU 開始型通信の場合のタスク（167 ページ）を参照してください。 • CSLU は CSSM から切断（製品インスタンス開始型通信および CSLU 開始型通信）については、トポロジのワークフロー：CSLU は CSSM から切断（171 ページ）を参照してください。 • SSM オンプレミス展開（製品インスタンス開始型通信）については、トポロジのワークフロー：SSM オンプレミス展開（176 ページ）を参照してください。
ステップ 5	**show license authorization** 例： `Device# show license authorization` `Overall status:` `  Active:` `PID:C9300X-24HX,SN:FOC2519L8R7` `  Status: SMART AUTHORIZATION INSTALLED on Oct 29 17:45:28 2021 UTC` `  Last Confirmation code: 6746c5b5` `  Standby:` `PID:C9300X-48HXN,SN:FOC2524L39P` `  Status: NOT INSTALLED` `  Member:` `PID:C9300X-48HX,SN:FOC2516LC92` `  Status: NOT INSTALLED` `Authorizations:` `  C9K HSEC (Cat9K HSEC):` `  Description: HSEC Key for Export Compliance on Cat9K Series Switches` `  Total available count: 1` `  Enforcement type: EXPORT RESTRICTED` `  Term information:` `  Active:` `PID:C9300X-24HX,SN:FOC2519L8R7` `  Authorization type: SMART`	製品インスタンスにインストールされている承認コードを表示します。

	コマンドまたはアクション	目的
	AUTHORIZATION INSTALLED License type: PERPETUAL Term Count: 1 Purchased Licenses: No Purchase Information Available	

CSSM からの SLAC の生成とファイルへのダウンロード

この手順を使用して、単一の製品インスタンスに対しても、複数の製品インスタンスに対しても SLAC を生成できます。

単一の製品インスタンスの場合、このタスクを実行するには PID とシリアル番号が必要です。製品インスタンスで、特権 EXEC モードで **show license udi** コマンドを入力し、情報を控えておきます。

複数の製品インスタンスの場合、該当するすべての製品インスタンスの PID とシリアル番号を含む .csv ファイルをアクセス可能な場所に保存します。

始める前に

サポートされるトポロジ：

- CSLU を介した CSSM への接続（製品インスタンス開始および CSLU 開始）
- CSLU は CSSM から切断（製品インスタンス開始および CSLU 開始）
- CSSM への接続なし、CSLU なし
- SSM オンプレミス展開（製品インスタンス開始型通信と SSM オンプレミス開始型通信）

手順

ステップ 1 https://software.cisco.com で CSSM Web UI にログインします。[Smart Software Licensing] で、[Manage licenses] リンクをクリックします。

シスコから提供されたユーザー名とパスワードを使用してログインします。

ステップ 2 [Inventory] タブをクリックします。

ステップ 3 [Virtual Account] ドロップダウンリストから、該当するバーチャルアカウントを選択します。

ステップ 4 [Product Instances] タブをクリックします。

ステップ 5 [Authorize License Enforced Features] タブをクリックします。

ステップ 6 単一の製品インスタンスまたは複数の製品インスタンスに SLAC を生成します（いずれかを選択）。

- **単一の製品インスタンスに SLAC を生成するには、次の手順を実行します。**

  1. [PID] と [Serial Number] を入力します。

（注）　他のフィールドは入力しないでください。

2. ライセンスを選択し、対応する [Reserve] 列に **1** を入力します。

   PID に対して正しいライセンスを選択したことを確認します。HSECK9 がサポートされている Cisco Catalyst アクセス、コア、およびアグリゲーションスイッチでは、[C9K HSEC] を選択します。

3. [Next] をクリックします。

4. [承認コードを生成（Generate Authorization Code）] をクリックします。

5. 承認コードをダウンロードし、.csv ファイルとして保存します。

6. 製品インスタンスへのファイルのインストール「製品インスタンスへのファイルのインストール （261 ページ）」を参照してください。

- **複数の製品インスタンスに SLAC を生成するには次の手順を実行します（この場合、.csv ファイルをアップロードしてください）。**

1. [Single Device]（デフォルト）というドロップダウンリストで、選択を [Multiple Devices] に変更します。

   この時点で、[Download a template] リンクが表示されます。必要なテンプレートまたはファイルがまだない場合は、ダウンロードできます。シリアル番号 PID のみが必須です。

2. [Choose File] をクリックし、SLAC を必要とする製品インスタンスのリストを含む .csv ファイルに移動します。

3. アップロードすると、デバイスのリストが CSSM に表示されます。すべてのデバイスのチェックボックスが有効になったら（すべてのデバイスの SLAC を要求することを意味します） [Next] をクリックします。

4. 各製品インスタンスに必要なライセンス数を指定し、[Next] をクリックします。

   （注）　「C9K HSEC」ライセンスの場合、UDI ごとに 1 つの SLAC が必要です。

5. [Reserve Licenses] をクリックします。

6. トポロジに従ってダウンロードします。

   - 「CSLU を介した CSSM への接続」、「CSLU は CSSM から切断」、「SSM オンプレミス展開」トポロジの場合は、[Download Authorization Codes] をクリックして、すべての承認コードを含む .csv ファイルをダウンロードします。[**閉じる（Close**）] をクリックします。

     これで、この .csv ファイルを CSLU または SSM オンプレミスにインポートできるようになりました。CSLU または SSM オンプレミスインターフェイスに戻り、残りの手順を実行してこのファイルをインポートします。

- 「CSM への接続なし、CSLU なし」トポロジ（外部との接続性がないネットワークで、コードを製品インスタンスにインポートする必要がある場合）では、各製品インスタンスの承認コードを別の .txt ファイルにダウンロードします。すべてのコードを含む .csv ファイルをダウンロードしないでください。

  CSSM Web UI で、[Inventory] > [Product Instances] タブに戻ります。各製品インスタンスを PID またはシリアル番号で検索します。UDI をクリックして、[Overview] タブを表示します。[Last Contact] フィールドに、[Download Reservation Authorization Code] というリンクが表示されます。リンクをクリックして、選択した製品インスタンスのみの承認コードを .txt 形式でダウンロードします。

  各 SLAC を製品インスタンスにインポートします。製品インスタンスへのファイルのインストール （261 ページ）を参照してください。

承認コードの返却

このタスクでは、ライセンスの承認コードを返却し、CSSM のライセンスプールにライセンスを返す方法を示します。この手順は、すべての承認コード（SLAC および SLR）に使用できます。

始める前に

サポートされるトポロジ：すべて

SLAC および SLR：返却するライセンスが使用中でないことを確認します。使用中の場合は、まず機能を無効に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	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。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 license summary** 例： `Device# show license summary` `License Usage:` `License Entitlement Tag Count Status` `---` `network-advantage (C9300-24 Network`	（任意）ライセンスの使用状況の概要を表示します。この手順は、SLAC を返却する場合にのみ適用されます。 暗号化機能を無効にした後でも、HSECK9 ライセンスのステータスが [IN USE] と表示される場合は、次の手順を実行します。この例の場合を示します。

	コマンドまたはアクション	目的
	Advan...) 1 IN USE dna-advantage (C9300-24 DNA Advantage) 1 IN USE network-advantage (C9300-48 Network Advan...) 2 IN USE dna-advantage (C9300-48 DNA Advantage) 2 IN USE **C9K HSEC (Cat9K HSEC) 1 IN USE**	HSECK9 ライセンスのステータスが [NOT IN USE] と表示された場合は、ステップ 5 に進みます。
ステップ 3	**platform hsec-license-release** 例： `Device# configure terminal` `Device(config)# platform hsec-license-release` `HSEC license is released` `Device(config)# exit`	（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、HSECK9 ライセンスを返却したら、特権 EXEC モードに戻ります。この手順は、SLAC を返却する場合にのみ適用されます。 HSECK9 ライセンスを使用する暗号化機能が無効または未設定で、ライセンスがまだ [IN USE] と表示されている場合、このコマンドにより強制的に HSECK9 ライセンスが [NOT IN USE] としてマークされます。
ステップ 4	**show license summary** 例： `Device# show license summary` `License Usage:` `License Entitlement Tag Count Status` `---` network-advantage (C9300-24 Network Advan...) 1 IN USE dna-advantage (C9300-24 DNA Advantage) 1 IN USE network-advantage (C9300-48 Network Advan...) 2 IN USE dna-advantage (C9300-48 DNA Advantage) 2 IN USE C9K HSEC (Cat9K HSEC) 0 NOT IN USE	返却するライセンスのステータスが [NOT IN USE] であることを確認します。使用中の場合は、まず機能を無効にする必要があります。
ステップ 5	**license smart authorization return{all | local}{offline[*path*] | online}** 例： `Device# license smart authorization return all online` `OR` `Device# license smart authorization return all offline` `Enter this return code in Cisco Smart`	CSSM のライセンスプールに承認コードを返却します。このコマンドを入力すると、戻りコードが表示されます。 製品インスタンスを指定します。 • all：高可用性セットアップまたはスタック構成セットアップで接続されたすべての製品インスタンスに対してアクションを実行します。

	コマンドまたはアクション	目的
	Software Manager portal: UDI: PID:C9300X-24HX,SN:FOC2519L8R7 Return code: CY9JHx-L1x5Rj-ftwzg1-h9QZAU-LE5DT1-babWeL-FABPt9-Wr1Dn7-Rp7 OR Device# **license smart authorization return all offline bootflash:return-code.txt**	• local：アクティブな製品インスタンスに対してアクションを実行します。これがデフォルトのオプションです。 CSSMに接続しているかどうかを指定します。 • CSSMに接続している場合、または製品インスタンス開始型通信のトポロジ（CSLUまたはSSMオンプレミス）を実装している場合は、**online**を入力します。コードは自動的にCSSMに返却され、確認が返されて製品インスタンスにインストールされます。このオプションを選択すると、戻りコードが自動的にCSSMに送信されます。 • CSSMに接続されていない場合、またはCSLU開始型通信またはSSMオンプレミス開始型通信のトポロジを実装した場合は、**offline**[*filepath_filename*]を入力します。**offline**キーワードのみを入力する場合は、CLIに表示される戻りコードをコピーし、CSSMに入力します。戻りコードをファイルに保存する場合は、ファイルからコードをコピーし、CSSMに同じコードを入力できます。ファイル形式は、読み取り可能な任意の形式にすることができます（これはアップロードされません）。例：`Device# license smart authorization return local offline bootflash:return-code.txt` • SLACを返却する場合は、次のタスクを実行してCSSMに戻りコードを入力します。CSSMでのSLAC戻りコードの入力と製品インスタンスの削除（255ページ） • SLR承認コードを返却する場合は、次のタスクを実行して

	コマンドまたはアクション	目的
		CSSMに戻りコードを入力します。CSSM での SLR 戻りコードの入力と製品インスタンスの削除（256ページ）この手順を完了してから、次の手順に進みます。
ステップ 6	**no license smart reservation** 例： `Device# configure terminal` `Device(config)# no license smart reservation` `Device(config)# exit`	グローバル コンフィギュレーション モードを開始し、製品インスタンスでSLR設定を無効化して、特権EXECモードに戻ります。 この手順は、返却する承認コードがSLR承認コードである場合にのみ必要です。返却するコードが HSECK9 ライセンスの SLAC である場合は、この手順をスキップします。 （注） この手順で **no license smart reservation** コマンドを入力する前に、オンラインまたはオフラインで承認コードの返却プロセス（**license smart authorization return**）を完了する必要があります。そうしないと、返却が CSSM または **show** コマンドに反映されない場合があります。問題を修正するには、シスコのテクニカルサポート担当者に連絡する必要があります。
ステップ 7	**show license authorization** 例： `Device# show license authorization` `License Authorizations` `=====================` `Overall status:` `  Active:` `PID:C9300X-24HX,SN:FOC2519L8R7` `  Status: NOT INSTALLED` `  Last return code:` `Cr9JHx-L1x5Rj-ftwzg1-h9QZAU-LE5DT1-babWeL-FABPt9-Wr1Dn7-Rp7` `  Standby:` `PID:C9300X-48HXN,SN:FOC2524L39P` `  Status: NOT INSTALLED`	ライセンス情報を表示します。出力の `License Authorizations` ヘッダーを確認します。返却プロセスが正常に完了すると、`Last return code:` フィールドに戻りコードが表示されます。

	コマンドまたはアクション	目的
	Member: PID:C9300X-48HX,SN:FOC2516LC92 Status: NOT INSTALLED <output truncated>	

CSSM での SLAC 戻りコードの入力と製品インスタンスの削除

このタスクを使用して、製品インスタンスが CSSM に接続されていない場合に、SLAC の返却手順を実行できます。これにより、ライセンスがライセンスプールに戻されます。さらに、CSSM から製品インスタンスを削除することもできます。

始める前に

サポートされるトポロジ：すべて

この手順は、SLAC を返却する場合にのみ実行してください。

承認コードの返却（251 ページ）に示すように、戻りコードが生成されていることを確認します。（このタスクの手順 7 で入力します）。

手順

ステップ 1 https://software.cisco.com で CSSM Web UI にログインします。[Smart Software Licensing] で、[Manage licenses] リンクをクリックします。

シスコから提供されたユーザー名とパスワードを使用してログインします。

ステップ 2 [Inventory] タブをクリックします。

ステップ 3 [Virtual Account] ドロップダウンリストから、バーチャルアカウントを選択します。

ステップ 4 [Product Instances] タブをクリックします。

使用可能な製品インスタンスのリストが表示されます。

ステップ 5 製品インスタンスリストから必要な製品インスタンスを見つけます。[Search] タブに PID またはシリアル番号を入力して検索できます。

ステップ 6 製品インスタンスの [Actions] 列で、[Actions] ドロップダウンリストから [Remove] を選択します。

[Remove Reservation] ウィンドウが表示されます。

ステップ 7 [Reservation Return Code] フィールドに、作成した SLAC 戻りコードを入力します。

ステップ 8 [Remove Reservation] をクリックします。

ライセンスがライセンスプールに戻されます。[Remove Reservation] ウィンドウが自動的に閉じ、[Product Instances] タブに戻ります。

（注） SLAC の返却のみの場合、これでタスクは終了です。CSSM から製品インスタンスも削除する場合は、次の手順に進みます。

ステップ 9 製品インスタンスの [Actions] 列で、[Actions] ドロップダウンリストから再度 [Remove] を選択します。

[Confirm Remove Product Instance] ウィンドウが表示されます。

ステップ 10 [Remove Product Instance] をクリックします。

製品インスタンスが CSSM から削除され、ライセンスが消費されなくなります。

CSSM での SLR 戻りコードの入力と製品インスタンスの削除

このタスクを使用して、SLR 承認コードの返却手順を実行できます。これによりライセンスがライセンスプールに返され、製品インスタンスが削除されます。

始める前に

サポートされるトポロジ：すべて

この手順は、SLR 承認コードを返す場合にのみ実行してください。

承認コードの返却（251 ページ）に示すように、戻りコードが生成されていることを確認します。（このタスクの手順 7 で入力します）。

手順

ステップ 1 https://software.cisco.com で CSSM Web UI にログインします。[Smart Software Licensing] で、[Manage licenses] リンクをクリックします。

シスコから提供されたユーザー名とパスワードを使用してログインします。

ステップ 2 [Inventory] タブをクリックします。

ステップ 3 [Virtual Account] ドロップダウンリストから、バーチャルアカウントを選択します。

ステップ 4 [Product Instances] タブをクリックします。

使用可能な製品インスタンスのリストが表示されます。

ステップ 5 製品インスタンスリストから必要な製品インスタンスを見つけます。[Search] タブに PID またはシリアル番号を入力して検索できます。

ステップ 6 製品インスタンスの [Actions] 列で、[Actions] ドロップダウンリストから [Remove] を選択します。

- 製品インスタンスが SLR 承認コードを含むライセンスを使用していない場合は、[Confirm Remove Product Instance] ウィンドウが表示されます。

- 製品インスタンスが SLR 承認コードを含むライセンスを使用している場合は、リターンコードを入力するためのフィールドのある [Remove Product Instance] ウィンドウが表示されます。

ステップ 7 [Reservation Return Code] フィールドに、作成したリターンコードを入力します。

（注） この手順は、製品インスタンスが SLR 承認コードを含むライセンスを使用している場合にのみ適用されます。

ステップ 8 [Remove Product Instance] をクリックします。

ライセンスがライセンスプールに返され、製品インスタンスが削除されます。

CSSM からの信頼コード用新規トークンの生成

信頼コードを要求するトークンを生成するには、次の手順を実行します。

所有するバーチャルアカウントごとに 1 つのトークンを生成します。1 つのバーチャルアカウントに属するすべての製品インスタンスに同じトークンを使用できます。

始める前に

サポートされるトポロジ：CSSM に直接接続

手順

ステップ 1 https://software.cisco.com で CSSM Web UI にログインします。[Smart Software Licensing] で、[Manage licenses] リンクをクリックします。

シスコから提供されたユーザー名とパスワードを使用してログインします。

ステップ 2 [Inventory] タブをクリックします。

ステップ 3 [Virtual Account] ドロップダウンリストから、必要なバーチャルアカウントを選択します。

ステップ 4 [General] タブをクリックします。

ステップ 5 [新規トークン（New Token）] をクリックします。[Create Registration Token] ウィンドウが表示されます。

ステップ 6 [Description] フィールドに、トークンの説明を入力します。

ステップ 7 [Expire After] フィールドに、トークンをアクティブにする必要がある日数を入力します。

ステップ 8 （オプション）[Max. Number of Uses] フィールドに、トークンの有効期限が切れるまでの最大使用回数を入力します。

ステップ 9 [Create Token] をクリックします。

ステップ 10 リストに新しいトークンが表示されます。[Actions] をクリックし、トークンを .txt ファイルとしてダウンロードします。

信頼コードのインストール

信頼コードを手動でインストールするには、次の手順を実行します。

始める前に

サポートされるトポロジ：

- CSSM に直接接続

手順

	コマンドまたはアクション	目的
ステップ 1	CSSMからの信頼コード用新規トークンの生成 （257 ページ）	まだ CSSM から信頼コードファイルを生成してダウンロードしていない場合は、生成とダウンロードを実行します。
ステップ 2	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。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 3	**license smart trust idtoken** *id_token_value* **{local | all} [force]** 例： `Device# license smart trust idtoken NGMwMjk5mYtNZaxMS00NzMZmtgWm all force`	CSSM との信頼できる接続を確立できます。*id_token_value* には、CSSM で生成したトークンを入力します。 次のいずれかのオプションを入力します。 • **local**：高可用性セットアップのアクティブデバイスに対してのみ信頼要求を送信します。これがデフォルトのオプションです。 • **all**：高可用性セットアップのすべてのデバイスに対して信頼要求を送信します。 製品インスタンスに既存の信頼コードがあるにもかかわらず、信頼コード要求を送信するには、**force** キーワードを入力します。

	コマンドまたはアクション	目的
		信頼コードは、製品インスタンスのUDIにノードロックされます。UDIがすでに登録されている場合、CSSMは同じUDIの新規登録を許可しません。**force** キーワードを入力すると、CSSMに送信されるメッセージに強制フラグが設定され、すでに存在する場合でも新しい信頼コードが作成されます。
ステップ 4	**show license status** 例： `<output truncated>` `Trust Code Installed:` `  Active:` `PID:C9500-24Y4C,SN:CAT2344L4GH` `  INSTALLED on Sep 04 01:01:46 2020 EDT` `  Standby:` `PID:C9500-24Y4C,SN:CAT2344L4GJ` `  INSTALLED on Sep 04 01:01:46 2020 EDT`	信頼コードがインストールされている場合は、日時が表示されます。日時はローカルタイムゾーンで表示されます。`Trust Code Installed:` フィールドを参照してください。

CSSM からのポリシーファイルのダウンロード

カスタムポリシーを要求した場合、または製品インスタンスに適用されるデフォルトとは異なるポリシーを適用する場合は、次のタスクを実行します。

始める前に

サポートされるトポロジ：

- CSSM への接続なし、CSLU なし
- CSLU は CSSM から切断

手順

ステップ 1 https://software.cisco.com で CSSM Web UI にログインします。[Smart Software Licensing] で、[Manage licenses] リンクをクリックします。

シスコから提供されたユーザー名とパスワードを使用してログインします。

ステップ 2 次のディレクトリパスを移動します。[Reports] > [Reporting Policy]。

ステップ 3 [Download] をクリックして、`.xml` ポリシーファイルを保存します。

これで、ファイルを製品インスタンスにインストールできます。「製品インスタンスへのファイルのインストール （261 ページ）」を参照してください。

CSSM への使用状況データのアップロードと ACK のダウンロード

製品インスタンスが CSSM や CSLU に接続されていない場合、または SSM オンプレミスが CSSM に接続されていない場合に RUM レポートを CSSM にアップロードして ACK をダウンロードするには、次のタスクを実行します。

始める前に

サポートされるトポロジ：

- CSSM への接続なし、CSLU なし
- SSM オンプレミス展開（製品インスタンス開始型通信と SSM オンプレミス開始型通信）

手順

ステップ 1 https://software.cisco.com で CSSM Web UI にログインします。[Smart Software Licensing] で、[Manage licenses] リンクをクリックします。

シスコから提供されたユーザー名とパスワードを使用してログインします。

ステップ 2 レポートを受信する**スマートアカウント**を選択します。

ステップ 3 [Smart Software Licensing] → [Reports] → [Usage Data Files] を選択します。

ステップ 4 [Upload Usage Data] をクリックします。ファイルの場所（tar 形式の RUM レポート）を参照して選択し、[Upload Data] をクリックします。

使用状況レポートは、アップロード後に CSSM で削除できません。

ステップ 5 [Select Virtual Accounts] ポップアップから、アップロードされたファイルを受信する**バーチャルアカウント**を選択します。ファイルが CSSM にアップロードされ、[Usage Data Files] タブエリアにファイル名、レポートの時刻、アップロード先のバーチャルアカウント、レポートステータス、レポートされた製品インスタンス数、確認ステータスとともに表示されます。

ステップ 6 [Acknowledgment] 列で [Download] をクリックして、アップロードしたレポートの `.txt` ACK ファイルを保存します。

[Acknowledgment] 列に「ACK」が表示されるまで待ちます。処理する RUM レポートが多数ある場合、CSSM では数分かかることがあります。

実装したトポロジに応じて、ファイルを製品インスタンスにインストールするか、またはCSLU に転送する、あるいは SSM オンプレミスにインポートすることができます。

製品インスタンスへのファイルのインストール

製品インスタンスにポリシーまたは ACK または SLAC をインポートしてインストールするには、次のタスクを実行します。

始める前に

サポートされるトポロジ：CSSM への接続なし、CSLU なし

対応するファイルは、製品インスタンスにアクセスできる場所に保存しています。

- ポリシーについては、CSSM からのポリシーファイルのダウンロード（259 ページ）を参照してください。
- ACK については、CSSM への使用状況データのアップロードと ACK のダウンロード（260 ページ）を参照してください。
- SLAC については、CSSM からの SLAC の生成とファイルへのダウンロード（249 ページ）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	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。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** *source* **bootflash:***file-name* 例： `Device# copy tftp://10.8.0.6/bootflash:example.txt`	（任意）ファイルをソースの場所またはディレクトリから製品インスタンスのフラッシュメモリにコピーします。また、リモートの場所からファイルを直接インポートし、製品インスタンスにインストールすることもできます（次の手順）。 • コピー元：これはファイルのコピー元の場所です。コピー元は、ローカルまたはリモートのいずれかです。 • **bootflash:**：これはブートフラッシュメモリの場合の宛先です。
ステップ 3	**license smart import** *filepath_file-name* 例： `Device# license smart import bootflash:example.txt`	ファイルを製品インスタンスにインポートしてインストールします。*filepath_filename* には、場所（ファイル名を含む）を指定します。インストール後、インストールしたファイルのタイプ

	コマンドまたはアクション	目的
		を示すシステムメッセージが表示されます。 （注） 複数の製品インスタンスに SLAC をインストールする場合（スタック設定など）、UDI ごとに個別の .txt SLAC ファイルをダウンロードしてください。一度に 1 つのファイルをインポートしてインストールします。
ステップ 4	**show license all** 例： `Device# show license all`	製品インスタンスのライセンス承認、ポリシー、およびレポート情報を表示します。

転送タイプ、URL、およびレポート間隔の設定

製品インスタンスの転送モードを設定するには、次のタスクを実行します。

始める前に

サポートされるトポロジ：すべて

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 モードを有効にします。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`	
ステップ 3	**license smart transport{automatic|callhome|cslu|off|smart}** 例： `Device(config)# license smart transport cslu`	使用する製品インスタンスの転送モードを設定します。次のオプションから選択します。 • **automatic**：転送モード **cslu** を設定します。 • **callhome**：転送モードとして Call Home を有効にします。

	コマンドまたはアクション	目的
		• **cslu**：これがデフォルトのトランスポートモードです。製品インスタンス開始型通信で CSLU または SSM オンプレミスを使用している場合は、このキーワードを入力します。 トランスポート モード キーワードは CSLU と SSM オンプレミスで同じですが、トランスポート URL は異なります。次の手順の **license smart url cslu** *cslu_or_on-prem_url* を参照してください。 • **off**：製品インスタンスからのすべての通信を無効にします。 • **smart**：スマート転送を有効にします。
ステップ 4	**license smart url**{*url* | **cslu** *cslu_url* | **default** | **smart** *smart_url* | **utility** *smart_url*} 例： `Device(config)# license smart url cslu` `http://192.168.0.1:8182/cslu/v1/pi`	設定された転送モードの URL を設定します。前の手順で選択した転送モードに応じて、対応する URL をここで設定します。 • *url*：転送モードとして **callhome** を設定している場合は、このオプションを設定します。CSSM URL を次のように正確に入力します。 `https://software.cisco.com/#module/SmartLicensing` **no license smart url** *url* コマンドは、デフォルトの URL に戻ります。 • **cslu** *cslu_or_on-prem_url*：トランスポートモードを **cslu** として設定している場合は、必要に応じて CSLU または SSM オンプレミスの URL を使用してこのオプションを設定します。 • CSLU を使用している場合は、次のように URL を入力します。 `http://<cslu_ip_or_host>:8182/cslu/v1/pi` `<cslu_ip_or_host>` には、CSLU をインストールした Windows

	コマンドまたはアクション	目的
		ホストのホスト名や IP アドレスを入力します。8182 はポート番号であり、CSLU が使用する唯一のポート番号です。 **no license smart url cslu** *cslu_url* コマンドは `http://cslu-local:8182/cslu/v1/pi` に戻ります • SSM オンプレミスを使用している場合は、次のように URL を入力します。 `http://<ip>/cslu/v1/pi/<tenant ID>` <ip> には、SSM オンプレミスをインストールしたサーバーのホスト名または IP アドレスを入力します。<tenantID> はデフォルトのローカル バーチャル アカウント ID にする必要があります。 **ヒント** SSM オンプレミスから URL 全体を取得できます。「トランスポート URL の取得（SSM オンプレミス UI）（234 ページ）」を参照してください **no license smart url cslu** *cslu_url* コマンドは `http://cslu-local:8182/cslu/v1/pi` に戻ります • **default**：設定されている転送モードによって異なります。このオプションでは、**smart** および **cslu** 転送モードのみがサポートされます。 転送モードが **cslu** に設定されている場合、**license smart url default** を設定すると、CSLU URL は自動的に

	コマンドまたはアクション	目的
		設定されます（https://cslu-local:8182/cslu/v1/pi）。 転送モードが **smart** に設定されている場合、**license smart url default** を設定すると、スマート URL は自動的に設定されます（https://smartreceiver.cisco.com/licservice/license）。 • **smart** *smart_url*：転送タイプとして **smart** を設定している場合は、このオプションを設定します。URL を次のように正確に入力します。 https://smartreceiver.cisco.com/licservice/license このオプションを設定すると、システムは **license smart url** *url* で自動的に URL の複製を作成します。重複するエントリは無視できます。これ以上の操作は必要ありません。 **no license smart url smart***smart_url* コマンドは、デフォルトの URL に戻ります。 • **utility** *smart_url*：このオプションは CLI では使用できますがサポートされていません。
ステップ 5	**license smart usage interval** *interval_in_days* 例： `Device(config)# license smart usage interval 40`	（任意）レポート間隔の日数を設定します。デフォルトでは、RUM レポートは 30 日ごとに送信されます。有効な値の範囲は 1 ～ 3650 です。 この値をゼロに設定すると、適用されるポリシーの指定内容に関係なく、RUM レポートは送信されません。これは、CSLU または CSSM が受信側にある可能性があるトポロジに適用されます。 ゼロより大きい値を設定し、通信タイプが**オフ**に設定されている場合、*interval_in_days* と `Ongoing reporting frequency(days):` のポリシー値の間で、値の小さい方が適用されます。たとえば、*interval_in_days* が 100 に設定され、ポリシーの値が `Ongoing reporting`

	コマンドまたはアクション	目的
		`frequency (days):90` の場合、RUM レポートは 90 日ごとに送信されます。 間隔を設定せず、デフォルトが有効な場合、レポート間隔は完全にポリシー値によって決定されます。たとえば、デフォルト値が有効で、不適用ライセンスのみが使用されている場合、ポリシーでレポートが不要と記述されていると、RUM レポートは送信されません。
ステップ 6	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	コンフィギュレーション ファイルに設定を保存します。

基本ライセンスまたはアドオンライセンスの設定

基本ライセンスまたはアドオンライセンスを注文および購入したら、使用する前にデバイスでライセンスを設定する必要があります。

このタスクではライセンスレベルを設定します。設定された変更を有効にする前にリロードが必要です。このタスクは、次の目的で使用できます。

- 現在のライセンスを変更する。
- 別のライセンスを追加する。たとえば、現在 Network Advantage を使用している場合、対応する Digital Networking Architecture（DNA）Advantage ライセンスで使用可能な機能も使用することができます。
- ライセンスを削除します。

始める前に

サポートされるトポロジ：すべて

使用可能な基本ライセンスとアドオンライセンスについては、基本ライセンスとアドオンライセンス （109 ページ）を参照してください。

購入したライセンスに関する情報は、Cisco Smart Software Manager（CSSM）Web UI の製品インスタンスのスマートアカウントとバーチャルアカウントで確認できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 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	**license boot level** *license_level* 例： `Device(config)# license boot level network-advantage add-on dna-advantage`	製品インスタンスで設定されたライセンスをアクティブにします。この例では、DNA Advantage ライセンスはリロード後に製品インスタンスでアクティブ化されます。
ステップ 4	**exit** 例： `Device(config)# exit`	特権 EXEC モードに戻ります。
ステップ 5	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	構成ファイルへの変更を保存します。
ステップ 6	**show version** 例： `Device# show version` `<output truncated>` `Technology Package License Information:` `------------------------------------------------` `Technology-package` `Technology-package` `Current Type` `Next reboot` `------------------------------------------------` `network-advantage Smart License` `network-advantage` `Subscription Smart License dna-advantage` `<output truncated>`	現在設定されているライセンスの情報と、リロード後に適用可能なライセンスを表示します。
ステップ 7	**reload** 例： `Device# reload`	デバイスがリロードされます。

次のタスク

ライセンスレベルを設定すると、変更はリロード後に有効になります。レポートが必要かどうかを確認するには、**show license status** 特権EXECコマンドの出力を参照し、`Next ACK deadline:` フィールドと `Next report push:` フィールドを確認します。

（注） ライセンスの使用状況の変更は、製品インスタンスに記録されます。レポートに関連した次の手順は、必要に応じて実行しますが、現在のトポロジによって異なります。

- CSLU を介して CSSM に接続
  - 製品インスタンス開始型通信：アクションは不要です。製品インスタンスは通信を開始すると、ポリシー（**show license status** —> [Next report push]）に従って、スケジュールされた時刻に最初の RUM レポートを自動的に CSLU に送信します。（製品インスタンスでこれを手動でトリガーするには、**license smart syncalllocal** 特権EXECコマンドを入力します。これにより、CSLU と製品インスタンスが同期され、保留中のデータが送受信されます）。CSLU は RUM レポートを CSSM に転送し、ACK を取得します。ACK は、製品インスタンスが次回 CSLU に接続したときに製品インスタンスに適用されます。
  - CSLU 開始型通信：CSLU インターフェイスで製品インスタンスから使用状況を収集します。使用状況レポートの収集：CSLU開始（CSLUインターフェイス） （214ページ） CSLUはRUM レポートをCSSMに送信し、CSSMからACKを取得します。CSLUが次に更新を実行するときに、ACK が製品インスタンスに適用されます。
- CSSM に直接接続：アクションは必要ありません。製品インスタンスは通信を開始すると、ポリシー（**show license status** —> [Next report push]）に従って、スケジュールされた時刻に最初の RUM レポートを自動的に CSSM に送信します。（製品インスタンスでこれを手動でトリガーするには、**license smart syncalllocal** 特権EXECコマンドを入力します。これにより、CSSMと製品インスタンスが同期され、保留中のデータが送受信されます）。ACK が使用可能になると、CSSM はこれを製品インスタンスに送り返します。
- CSLU は CSSM から切断
  - 製品インスタンス開始型通信：アクションは不要です。製品インスタンスは通信を開始すると、ポリシー（**show license status** —> [Next report push]）に従って、スケジュールされた時刻に最初の RUM レポートを自動的に CSLU に送信します。（製品インスタンスでこれを手動でトリガーするには、**license smart syncalllocal** 特権EXECコマンドを入力します。これにより、CSLU と製品インスタンスが同期され、保留中のデータが送受信されます）。

    CSLU が CSSM から切断されているため、CSLU インターフェイスと CSSM Web UI でタスク CSSMへのエクスポート（CSLUインターフェイス） （215ページ）>CSSMへの使用状況データのアップロードと ACK のダウンロード （260 ページ）>CSSMからのインポート（CSLU インターフェイス） （216 ページ） を実行します。ACK は、製品インスタンスが次回CSLUに接続したときに製品インスタンスに適用されます。

- CSLU 開始型通信：CSLU インターフェイスで製品インスタンスから使用状況を収集します。使用状況レポートの収集：CSLU 開始（CSLU インターフェイス） （214 ページ）

  CSLU が CSSM から切断されているため、CSLU インターフェイスと CSSM Web UI でタスク CSSM へのエクスポート（CSLU インターフェイス） （215 ページ） > CSSM への使用状況データのアップロードと ACK のダウンロード （260 ページ） > CSSM からのインポート（CSLU インターフェイス） （216 ページ） を実行します。CSLU が次に更新を実行するときに、ACK が製品インスタンスに適用されます。

- コントローラを介して CSSM に接続：アクションは必要ありません（Cisco DNA Center GUI で最初のアドホックレポートをすでに完了している場合）。Cisco DNA Center は、後続のすべてのレポートを処理し、製品インスタンスに ACK を返します。

- CSSM への接続なし、CSLU なし：RUM レポートを（製品インスタンスの）ファイルに保存してから、CSSM にアップロードします（インターネットとシスコに接続されているワークステーションから）。**license smart save usage** コマンドを特権 EXEC モードで実行し、RUM レポートをファイルに保存します。次に、CSSM にファイルをアップロードして ACK をダウンロードするため、次のタスクを実行します。CSSM への使用状況データのアップロードと ACK のダウンロード （260 ページ）最後に、製品インスタンスに ACK をインストールするため、次のタスクを実行します。製品インスタンスへのファイルのインストール （261 ページ）

- SSM オンプレミス展開：

  - 製品インスタンス開始型通信：アクションは不要です。製品インスタンスは通信を開始すると、ポリシー（**show license status** —> [Next report push]）に従って、スケジュールされた時刻に最初の RUM レポートを自動的に SSM オンプレミスに送信します。（製品インスタンスでこれを手動でトリガーするには、**license smart syncalllocal** 特権 EXECコマンドを入力します。これにより、SSM オンプレミスと製品インスタンスが同期され、保留中のデータが送受信されます）。

    - SSM オンプレミスが CSSM に接続されている場合、SSM オンプレミス インターフェイスで、[Reports] > [Usage Schedules] > [Synchronization schedule with Cisco] に移動します。

    - SSM オンプレミスが CSSM から切断されている場合は、レポートに必要なファイルをアップロードおよびダウンロードします。使用状況データのエクスポートとインポート（SSM オンプレミス UI） （235 ページ）

  - SSM オンプレミス開始型通信：SSM オンプレミス インターフェイスで、製品インスタンスから使用状況情報を収集します。[Reports] > [Synchronisation pull schedule with the devices] > [Synchronize now with the device] に移動します。

    - SSM オンプレミスが CSSM に接続されている場合、SSM オンプレミス インターフェイスで、[Reports] > [Usage Schedules] > [Synchronization schedule with Cisco] に移動します。

- SSM オンプレミスが CSSM から切断されている場合は、レポートに必要なファイルをアップロードおよびダウンロードします。使用状況データのエクスポートとインポート（SSM オンプレミス UI）（235 ページ）

リソース使用率測定レポートの例

次に、リソース使用率測定（RUM）レポートの例を XML 形式で示します（RUM レポートおよびレポート確認応答（146 ページ）を参照）。このような複数のレポートを連結して 1 つのレポートを形成できます。

```
<?xml version="1.0" encoding="UTF-8"?>
  <smartLicense>
```

[redacted]

```
  </smartLicense>
```

ポリシーを使用したスマートライセンシングのトラブルシューティング

このセクションでは、発生する可能性のあるポリシーを使用したスマートライセンシングに関連するシステムメッセージ、考えられる失敗の理由、および推奨するアクションを示します。

システム メッセージの概要

システムメッセージは、システムソフトウェアからコンソール（および任意で別のシステムのロギングサーバー）に送信されます。すべてのシステム メッセージがシステムの問題を示すわけではありません。通知目的のメッセージもあれば、通信回線、内蔵ハードウェア、またはシステム ソフトウェアの問題を診断するうえで役立つメッセージもあります。

システム メッセージの読み方

システムログメッセージには最大 80 文字を含めることができます。各システム メッセージはパーセント記号（%）から始まります。構成は次のとおりです。

```
%FACILITY-SEVERITY-MNEMONIC: Message-text
```

%FACILITY

メッセージが参照するファシリティを示す 2 文字以上の大文字です。ファシリティは、ハードウェアデバイス、プロトコル、またはシステムソフトウェアのモジュールなどです。

SEVERITY

0 ～ 7 の 1 桁のコードで、状態の重大度を表します。この値が小さいほど、重大な状況を意味します。

表 *18* : メッセージの重大度

重大度	説明
0：緊急	システムが使用不可能な状態。
1：アラート	ただちに対応が必要な状態。
2：クリティカル	危険な状態。
3：エラー	エラー条件。
4：警告	警告条件。
5：通知	正常だが注意を要する状態。
6：情報	情報メッセージのみ。
7：デバッグ	デバッグ時に限り表示されるメッセージのみ。

MNEMONIC

メッセージを一意に識別するコード。

Message-text

メッセージテキストは、状態を説明したテキスト文字列です。メッセージのこの部分には、端末ポート番号、ネットワーク アドレス、またはシステム メモリ アドレス空間の位置に対応するアドレスなど、イベントの詳細情報が含まれることがあります。この可変フィールドの情報はメッセージごとに異なるので、ここでは角カッコ（[]）で囲んだ短い文字列で示します。たとえば 10 進数は [dec] で表します。

表 *19* : メッセージの変数フィールド

重大度	説明
[char]	1 文字
[chars]	文字列
[dec]	10 進数
[enet]	イーサネット アドレス（たとえば 0000.FEED.00C0）
[hex]	16 進数
[inet]	インターネット アドレス（10.0.2.16）
[int]	整数

重大度	説明
[node]	アドレス名またはノード名
[t-line]	8 進数のターミナルライン番号（10 進数 TTY サービスが有効な場合は 10 進数）
[clock]	クロック（例：01:20:08 UTC Tue Mar 2 1993）

システム メッセージ

このセクションでは、発生する可能性のあるポリシーを使用したスマートライセンシングに関連するシステムメッセージ、考えられる失敗の理由（失敗メッセージの場合）、および推奨するアクション（アクションが必要な場合）を示します。

すべてのエラーメッセージについて、問題を解決できない場合は、シスコのテクニカルサポート担当者に次の情報をお知らせください。

コンソールまたはシステムログに出力されたとおりのメッセージ。

show license tech support、**show license history message**、および **show platform software sl-infra** 特権 EXEC コマンドの出力。

ポリシーを使用したスマートライセンシング関連のシステムメッセージ：

- %SMART_LIC-3-POLICY_INSTALL_FAILED
- %SMART_LIC-3-AUTHORIZATION_INSTALL_FAILED
- %SMART_LIC-3-COMM_FAILED
- %SMART_LIC-3-COMM_RESTORED
- %SMART_LIC-3-POLICY_REMOVED
- %SMART_LIC-3-TRUST_CODE_INSTALL_FAILED
- %SMART_LIC-4-REPORTING_NOT_SUPPORTED
- %SMART_LIC-6-POLICY_INSTALL_SUCCESS
- %SMART_LIC-6-AUTHORIZATION_INSTALL_SUCCESS
- %SMART_LIC-6-AUTHORIZATION_REMOVED
- %SMART_LIC-6-REPORTING_REQUIRED
- %SMART_LIC-6-TRUST_CODE_INSTALL_SUCCESS

```
Error Message %SMART_LIC-3-POLICY_INSTALL_FAILED: The installation of a new
licensing policy has failed: [chars].
```

説明：ポリシーがインストールされましたが、ポリシーコードの解析中にエラーが検出され、インストールに失敗しました。[chars] はエラーの詳細を示すエラー文字列です。

失敗の理由として次が考えられます。

- 署名の不一致：これは、システムクロックが正確でないことを意味します。
- タイムスタンプの不一致：製品インスタンスのシステムクロックがCSSMと同期していないことを意味します。

推奨するアクション：

考えられる両方の失敗の理由に関しては、システムクロックが正確で、CSSMと同期していることを確認します。**ntp server** コマンドをグローバル コンフィギュレーション モードで設定します。次に例を示します。

```
Device(config)# ntp server 198.51.100.100 version 2 prefer
```

前述の手順を実行しても、ポリシーのインストールが失敗する場合は、シスコのテクニカルサポート担当者にお問い合わせください。

```
Error Message %SMART_LIC-3-AUTHORIZATION_INSTALL_FAILED: The install of a new
licensing authorization code has failed on [chars]: [chars].
```

説明：承認コードのインストールを試みましたが、インストールに失敗しました。最初の[chars]は承認コードのインストールが失敗したUDI、2番目の[chars]はエラーの詳細を示すエラー文字列です。

失敗の理由として次が考えられます。

- 現在設定されている機能の承認に必要な十分なライセンスがない：これは、必要な数の承認コードを提供していなかったことを意味します。
- UDIの不一致：承認コードファイル内の1つ以上のUDIが、承認コードファイルをインストールする製品インスタンスと一致していません。複数のUDIの承認コードを生成した場合、高可用性またはスタック構成セットアップでは、承認コードファイルにリストされているすべてのUDIが、高可用性またはスタック構成セットアップのすべてのUDIと一致する必要があります。一致しない場合、インストールは失敗します。

  承認コードファイル内のすべてのUDIを製品インスタンスのUDI（スタンドアロンまたは高可用性）と照合します。

  ```
  Excerpt of UDI information in a SLAC file:
  <smartLicenseAuthorization>
  <udi>P:C9300X-24HX,SN:FOC2519L8R7</udi>

  <output truncated>
  </smartLicenseAuthorization>

  Sample output of UDI information on a product instance:
  Device# show license udi
  UDI: PID:C9300X-24HX,SN:FOC2519L8R7
  ```

- 署名の不一致：これは、システムクロックが正確でないことを意味します。クロックが同期されていない場合、SLACの要求時の試行は**show license tech**の出力に反映されません。

  ```
  Authorization Confirmation:
    Attempts: Total=0, Success=0, Fail=0  Ongoing Failure: Overall=0 Communication=0
  ```

推奨処置

- **show license tech support** コマンドの出力で、`Failure Reason:` フィールドを確認し、失敗した理由を確認します。

```
Device# show license tech support
<output truncated>

Communication Statistics:
=========================
Authorization Confirmation:
  Attempts: Total=2, Success=2, Fail=0  Ongoing Failure: Overall=0 Communication=0
  Last Response: OK on Sep 23 17:51:52 2020 UTC
    Failure Reason: <none>
  Last Success Time: Sep 23 17:51:52 2020 UTC
  Last Failure Time: <none>
```

- 現在設定されている機能の承認に必要な十分なライセンスがない、およびUDIの不一致：

- **show license udi** コマンドを使用して、UID の正しい完全なリストがあることを確認します。このコマンドは、高可用性およびスタック構成セットアップの場合にすべての製品インスタンスを表示します。その後、SLAC を再度インストールします。

- 署名の不一致：システムクロックが正確で、CSSMと同期していることを確認します。確認するためには、グローバル コンフィギュレーション モードで **ntp server** コマンドを設定します。次に例を示します。

```
Device(config)# ntp server 198.51.100.100 version 2 prefer
```

この設定が完了したら、再度 **show license tech** を使用してクロックが実際に同期されているかどうかを確認します。正常に同期されると、[Clock sync-ed with NTP] フィールドが [True] に設定されます。同期されていない場合、このフィールドは [False] に設定されます。

```
Error Message %SMART_LIC-3-COMM_FAILED: Communications failure with the [chars] :
[chars]
```

説明：CSSM、CSLU、または SSM オンプレミスのいずれかとのスマートライセンシング通信が失敗しました。最初の [chars] は現在設定されている転送タイプで、2 番めの [chars] はエラーの詳細を示すエラー文字列です。このメッセージは、失敗した通信の試行ごとに表示されます。

失敗の理由として次が考えられます。

- CSSM、CSLU、または SSM オンプレミスに到達できない：これは、ネットワーク到達可能性に問題があることを意味します。

- 404 ホストが見つからない：これは CSSM サーバーがダウンしていることを意味します。

正インスタンスが RUM レポートの送信を開始するトポロジ（CSLU を介して CSSM に接続：製品インスタンス開始型通信、CSSM から切断されているCSSM、CSLU への直接接続：製品スタンス開始型通信、および SSM オンプレミス展開：製品インスタンス開始型通信）では、

この通信障害メッセージがスケジュールされたレポート（**license smart usage interval** *interval_in_days* グローバル コンフィギュレーション コマンド）と一致している場合は、製品インスタンスはスケジュールされた時間が経過した後、最大 4 時間にわたって RUM レポートを送信しようとします。（通信障害が続くために）それでもレポートを送信できない場合、システムは間隔を 15 分にリセットします。通信障害が解消されると、レポート間隔は最後に設定された値に戻ります。

推奨するアクション：

CSSMに到達できない場合、CSLUに到達できない場合、およびSSMオンプレミスに到達できない場合のトラブルシューティング手順を示します。

CSSM が到達不能で、設定されている転送タイプが smart の場合：

1. スマート URL が正しく設定されているかどうかを確認します。特権 EXEC モードで **show license status** コマンドを使用して、URL が次のようになっているかどうかを確認します。https://smartreceiver.cisco.com/licservice/licenseそうでない場合は、グローバル コンフィギュレーション モードで **license smart url smart** *smar_URL* コマンドを再設定します。

2. DNS 解決を確認します。製品インスタンスが `smartreceiver.cisco.com` または nslookup で変換された IP に対して ping を実行できることを確認します。次の例は、変換された IP に対して ping を実行する方法を示しています。

```
Device# ping 171.70.168.183
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 171.70.168.183, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
```

CSSM が到達不能で、設定されている転送タイプが callhome の場合：

1. URL が正しく入力されているかどうかを確認します。特権 EXEC モードで **show license status** コマンドを使用して、URL が次のようになっているかどうかを確認します。https://tools.cisco.com/its/service/oddce/services/DDCEService

2. Call Home プロファイル `CiscoTAC-1` がアクティブで、接続先 URL が正しいことを確認します。**show call-home profile all** コマンドは特権 EXEC モードで使用してください。

```
Current smart-licensing transport settings:
 Smart-license messages: enabled
 Profile: CiscoTAC-1 (status: ACTIVE)
 Destination  URL(s):  https://tools.cisco.com/its/service/oddce/services/DDCEService
```

3. DNS 解決を確認します。製品インスタンスが `tools.cisco.com` または nslookup で変換された IP に対して ping を実行できることを確認します。

```
Device# ping tools.cisco.com
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 173.37.145.8, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 41/41/42 ms
```

上記の方法で解決しない場合は、製品インスタンスが設定されているかどうか、製品インスタンスのIPネットワークが稼働しているかどうかを確認します。ネットワークが稼働し

ていることを確認するには、インターフェイス コンフィギュレーション モードで **no shutdown** コマンドを設定します。

デバイスがサブネット IP でサブネットマスクされているかどうか、および DNS IP が設定されているかどうかを確認します。

4. HTTPS クライアントの送信元インターフェイスが正しいことを確認します。

   現在の設定を表示するには、特権 EXEC モードで **show ip http client** コマンドを使用します。グローバル コンフィギュレーション モードで **ip http client source-interface** コマンドを使用して、再設定します。

   上記の方法で解決しない場合は、ルーティングルール、およびファイアウォール設定を再確認します。

CSLU に到達できない場合：

1. CSLU 検出が機能するかどうかを確認します。

   - `cslu-local` のゼロタッチ DNS 検出またはドメインの DNS 検出。

     show license all コマンドの出力で、`Last ACK received:` フィールドを確認します。このフィールドに最新のタイムスタンプがある場合は、製品インスタンスが CSLU と接続されていることを意味します。ない場合は、次のチェックに進みます。

     製品インスタンスが `cslu-local` に対して ping を実行できるかどうかを確認します。ping が成功すると、製品インスタンスが到達可能であることが確認されます。

     上記の方法で解決しない場合は、ホスト名 `cslu-local` が CSLU の IP アドレス（CSLU をインストールした Windows ホスト）にマッピングされているエントリを使用してネームサーバーを設定します。グローバル コンフィギュレーション モードで **ip domain name** *domain-name* コマンドと **ip name-server** *server-address* コマンドを設定します。この例では、CSLU IP は 192.168.0.1 で、name-server によってエントリ `cslu-local.example.com` が作成されます。

     ```
     Device(config)# ip domain name example.com
     Device(config)# ip name-server 192.168.0.1
     ```

   - CSLU URL が設定されています。

     show license all コマンド出力の `Transport:` ヘッダーで、次の点を確認します。 `Type:` は `cslu` で、`Cslu address:` は CSLU をインストールした Windows ホストのホスト名または IP アドレスになっている必要があります。残りのアドレスが下記のように設定されているかどうかを確認するとともに、ポート番号が 8182 であるかどうかを確認します。

     ```
     Transport:
       Type: cslu
       Cslu address: http://192.168.0.1:8182/cslu/v1/pi
     ```

     そうでない場合は、グローバル コンフィギュレーション モードで **license smart transport cslu** および **license smart url cslu** *http://<cslu_ip_or_host>:8182/cslu/v1/pi* コマンドを設定します。

2. CSLU 開始型通信の場合、上記の CSLU 検出チェックに加えて、次の点を確認します。

   HTTP 接続を確認します。特権 EXEC モードで **show ip http server session-module** コマンドを使用します。出力の `HTTP server current connections:` ヘッダーで、`SL_HTTP` がアクティブになっていることを確認します。CSLU 開始型通信のネットワーク到達可能性の確認（216 ページ） で説明されているとおりに **ip http** が再設定されていない場合：

   CSLU がインストールされているデバイスの Web ブラウザで、`https://<product-instance-ip>/` を確認します。これにより、CSLU から製品インスタンスへの REST API が期待どおりに動作することが保証されます。

SSM オンプレミスに到達できない場合：

1. 製品インスタンス開始型通信の場合は、SSM オンプレミスのトランスポートタイプと URL が正しく設定されているかどうかを確認します。

   show license all コマンドの出力の `Transport:` ヘッダーの下で、`Type:` が `cslu` であり、`Cslu address:` には、SSM オンプレミスにインストールしたサーバーのホスト名または IP アドレスと、デフォルトのローカル バーチャル アカウントの <tenantID> があることを確認します。次の例を参照してください。

   ```
   Transport:
     Type: cslu
     Cslu address: https://192.168.0.1/cslu/v1/pi/on-prem-default
   ```

   SSM オンプレミスの正しい URL があることを確認し（トランスポート URL の取得（SSM オンプレミス UI）（234 ページ）を参照）、次に、グローバル コンフィギュレーション モードで **license smart transport cslu** コマンドと **license smart url cslu** *http://<ip>/cslu/v1/pi/<tenant ID>* コマンドを設定します。

   製品インスタンス開始型通信のネットワーク到達可能性の確認（232 ページ）に記載されているように、ネットワークに必要な他のコマンドが設定されていることを確認します。

2. SSM オンプレミス開始型通信の場合は、HTTPS 接続を確認します。

   特権 EXEC モードで **show ip http server session-module** コマンドを使用します。出力の `HTTP server current connections:` ヘッダーで、`SL_HTTP` がアクティブになっていることを確認します。SSM オンプレミス開始型通信のネットワーク到達可能性の確保（237 ページ） で説明されているとおりに **ip http** コマンドが再設定されていない場合は、次の手順を実行します。

3. トラストポイントと証明書が受け入れられることを確認します。

   SSM オンプレミス展開の両方の通信形式で、正しいトラストポイントが使用され、必要な証明書が受け入れられることを確認します。

   ```
   Device(config)# crypto pki trustpoint SLA-TrustPoint
   Device(ca-trustpoint)#
   Device(ca-trustpoint)# enrollment terminal
   Device(ca-trustpoint)# revocation-check none
   Device(ca-trustpoint)# end
   Device# copy running-config startup-config
   ```

上記がうまくいかず、通信障害が続く場合は、シスコのテクニカルサポート担当者にお問い合わせください。

```
Error Message  %SMART_LIC-3-COMM_RESTORED: Communications with the [chars] restored.
[chars] - depends on the transport type
        - Cisco Smart Software Manager (CSSM)
        - Cisco Smart License utility (CSLU)
Smart Agent communication with either the Cisco Smart Software Manager (CSSM) or the
Cisco Smart License
utility (CSLU) has been restored. No action required.
```

説明：CSSM、CSLU、または SSM オンプレミスのいずれかとの製品インスタンス通信が復元されます。

推奨するアクション：アクションは必要ありません。

```
Error Message %SMART_LIC-3-POLICY_REMOVED: The licensing policy has been removed.
```

説明：以前にインストールしたカスタムライセンスポリシーが削除されました。`Cisco default` ポリシーが自動的に有効になります。これにより、スマートライセンシングの動作が変更される可能性があります。

失敗の理由として次が考えられます。

特権 EXEC モードで **license smart factory reset** コマンドを入力すると、ポリシーを含むすべてのライセンス情報が削除されます。

推奨するアクション：

ポリシーが意図的に削除された場合、それ以上のアクションは不要です。

ポリシーが誤って削除された場合は、ポリシーを再適用できます。実装したトポロジに応じて、該当するメソッドに従ってポリシーを取得します。

- CSSM に直接接続：

  show license status を入力し、`Trust Code Installed:` フィールドを確認します。信頼が確立されると、CSSMは再度ポリシーを自動的に返します。ポリシーは、対応するバーチャルアカウントのすべての製品インスタンスに自動的に再インストールされます。

  信頼が確立されていない場合は、次のタスクを実行します。CSSMからの信頼コード用新規トークンの生成（257ページ）および信頼コードのインストール（258ページ）これらのタスクを完了すると、CSSMは再度ポリシーを自動的に返します。その後、バーチャルアカウントのすべての製品インスタンスにポリシーが自動的にインストールされます。

- CSLU を介して CSSM に接続：

- 製品インスタンス開始型通信の場合は、特権 EXEC モードで **license smart sync** コマンドを入力します。同期要求により、CSLU は欠落している情報（ポリシーまたは承認コード）を製品インスタンスにプッシュします。
- CSLU 開始型通信の場合は、次のタスクを実行します。使用状況レポートの収集：CSLU開始（CSLUインターフェイス）（214ページ）タスクを実行すると、CSLUはACK 応答で欠落しているポリシーを検出して再提供します。

• CSLU は CSSM から切断：

- 製品インスタンス開始型通信の場合は、特権 EXEC モードで **license smart sync** コマンドを入力します。同期要求により、CSLU は欠落している情報（ポリシーまたは承認コード）を製品インスタンスにプッシュします。次に、次のタスクを指定された順序で実行します。CSSMへのエクスポート（CSLUインターフェイス）（215ページ）>CSSM への使用状況データのアップロードと ACK のダウンロード（260 ページ）> CSSM からのインポート（CSLU インターフェイス）（216 ページ）
- CSLU 開始型通信の場合は、次のタスクを実行します。使用状況レポートの収集：CSLU開始（CSLUインターフェイス）（214ページ）タスクを実行すると、CSLUはACK 応答で欠落しているポリシーを検出して再提供します。次に、次のタスクを指定された順序で実行します。CSSM へのエクスポート（CSLU インターフェイス）（215 ページ）>CSSM への使用状況データのアップロードと ACK のダウンロード（260ページ）>CSSMからのインポート（CSLUインターフェイス）（216ページ）

• CSSM への接続なし、CSLU なし

完全に外部との接続性がないネットワークにいる場合は、インターネットとCSSMに接続できるワークステーションから次のタスク：CSSMからのポリシーファイルのダウンロード（259ページ）および製品インスタンスへのファイルのインストール（261ページ）を実行します。

• SSM オンプレミス展開

- 製品インスタンス開始型通信の場合は、特権 EXEC モードで **license smart sync** コマンドを入力します。製品インスタンスを SSM オンプレミスと同期させ、必要な情報または欠落している情報を復元する原因です。必要に応じて、SSM オンプレミスと CSSM を同期します。
- SSM オンプレミス開始型通信の場合：SSM オンプレミス UI で、[Reports] > [Synchronization pull schedule] > [Synchronize now with the device] に移動します。

SSMオンプレミス展開の両方の通信形式で、次のいずれかのオプションを使用してCSSMと同期します。

- SSM オンプレミスが CSSM に接続されている場合：SSM オンプレミス UI の [Smart Licensing] ワークスペースで、[Reports] > [Usage Schedules] > [Synchronize now with Cisco] に移動します。

- SSM オンプレミスが CSSM に接続されていません。使用状況データのエクスポートとインポート（SSM オンプレミス UI）（235 ページ）を参照してください。

```
Error Message %SMART_LIC-3-TRUST_CODE_INSTALL_FAILED: The install of a new licensing
trust code has failed on [chars]: [chars].
```

説明：信頼コードのインストールに失敗しました。最初の [chars] は、信頼コードのインストールが試行された UDI です。2 番目の [chars] は、エラーの詳細を示すエラー文字列です。

失敗の理由として次が考えられます。

- 信頼コードがすでにインストールされています。信頼コードは製品インスタンスの UDI にノードロックされています。UDI がすでに登録されている場合に別の UDI をインストールしようとすると、インストールは失敗します。
- スマートアカウントとバーチャルアカウントの不一致：これは、（トークン ID が生成された）スマートアカウントまたはバーチャルアカウントに、信頼コードをインストールした製品インスタンスが含まれていないことを意味します。CSSM で生成されたトークンは、スマートアカウントまたはバーチャルアカウントレベルで適用され、そのアカウントのすべての製品インスタンスにのみ適用されます。
- 署名の不一致：これは、システムクロックが正確でないことを意味します。
- タイムスタンプの不一致：製品インスタンスの時刻が CSSM と同期していないため、インストールが失敗する可能性があります。

推奨するアクション：

- 信頼コードはすでにインストールされています。製品インスタンスに信頼コードがすでに存在する状況で信頼コードをインストールする場合は、特権 EXEC モードで **license smart trust idtoken** *id_token_value* **{local | all} [force]** コマンドを再設定します。再設定の際、**force** キーワードを必ず含めてください。**force** キーワードを入力すると、CSSM に送信されるメッセージに強制フラグが設定され、すでに存在する場合でも新しい信頼コードが作成されます。
- スマートアカウントとバーチャルアカウントの不一致：

  https://software.cisco.com で CSSM Web UI にログインします。[Smart Software Licensing] で、[Manage licenses] リンクをクリックします。[Inventory] タブをクリックします。[Virtual Account] ドロップダウンリストから、必要なバーチャルアカウントを選択します。[Product Instances] タブをクリックします。

  トークンを生成する製品インスタンスが、選択したバーチャルアカウントにリストされているかどうかを確認します。その場合は、次の手順：CSSM からの信頼コード用新規トークンの生成（257 ページ）および信頼コードのインストール（258 ページ）に進みます。リストされていない場合は、正しいスマートアカウントとバーチャルアカウントを確認して選択します。その後、次の手順を実行します。

- タイムスタンプの不一致と署名の不一致：グローバル コンフィギュレーション モードで **ntp server** コマンドを設定します。次に例を示します。

  ```
  Device(config)# ntp server 198.51.100.100 version 2 prefer
  ```

```
Error Message    %SMART_LIC-4-REPORTING_NOT_SUPPORTED: The CSSM OnPrem that this
product instance is connected to is down rev and does not support the enhanced policy
and usage
reporting mode.
```

説明：Cisco Smart Software Manager オンプレミス（旧称 Cisco Smart Software Manager サテライト）は、Cisco IOS XE Amsterdam 17.3.3 以降でのみ Smart Licensing Using Policy 環境でサポートされています（SSM オンプレミス （142 ページ）を参照）。サポートされていないリリースでは、製品インスタンスは次のように動作します。

- 登録の更新と承認の更新の送信を停止します。
- 使用状況の記録を開始し、RUM レポートをローカルに保存します。

推奨するアクション：

次の選択肢があります。

- 代わりに、サポートされているトポロジを参照し、いずれかを実装します。サポートされるトポロジ （147 ページ）を参照してください。
- Smart Licensing Using Policy で SSM オンプレミスがサポートされているリリースにアップグレードします。Smart Licensing Using Policy をサポートする SSM オンプレミスのバージョンへの移行 （208 ページ）を参照してください。

```
Error Message %SMART_LIC-6-POLICY_INSTALL_SUCCESS: A new licensing policy
was successfully installed.
```

説明：次のいずれかの方法でポリシーがインストールされました。

- Cisco IOS コマンドの使用
- CSLU 開始型通信
- ACK 応答の一部として

推奨するアクション：アクションは必要ありません。適用されているポリシー（使用中のポリシー）とそのレポート要件を確認するには、特権 EXEC モードで **show license all** コマンドを入力します。

```
Error Message %SMART_LIC-6-AUTHORIZATION_INSTALL_SUCCESS: A new licensing
authorization code was successfully installed on: [chars].
```

説明： [chars] は、承認コードが正常にインストールされた UDI です。

推奨するアクション：アクションは必要ありません。インストールされた承認コードの詳細を確認するには、特権 EXEC モードで **show license authorization** コマンドを入力します。


```
Error Message %SMART_LIC-6-AUTHORIZATION_REMOVED: A licensing authorization code has
 been removed from [chars]
```

説明： [chars] は、承認コードがインストールされた UDI です。承認コードが削除されました。これにより、製品インスタンスからライセンスが削除され、スマートライセンシングとライセンスを使用する機能の動作が変更される可能性があります。

推奨するアクション：アクションは必要ありません。ライセンスの現在の状態を確認するには、特権 EXEC モードで **show license all** コマンドを入力します。


```
Error Message %SMART_LIC-6-REPORTING_REQUIRED: A Usage report acknowledgement
will be required in [dec] days.
```

説明：これは、シスコへの RUM レポートが必要であることを意味するアラートです。[dec] は、このレポート要件を満たすために残された時間（日数）です。

推奨するアクション：要求された時間内に RUM レポートが送信されるようにします。実装したトポロジによって、レポート方式が決まります。

- CSLU を介して CSSM に接続
  - 製品インスタンス開始型通信の場合：特権 EXEC モードで **license smart sync** コマンドを入力します。CSLU が現在 CSSM にログインしている場合、CSSM 内の関連付けられているスマートアカウントとバーチャルアカウントに自動的に送信されます。
  - CSLU 開始型通信の場合は、次のタスクを実行します。使用状況レポートの収集：CSLU 開始（CSLU インターフェイス） （214 ページ）
- CSSM への直接接続：特権 EXEC モードで **license smart sync** コマンドを入力します。
- コントローラを介して CSSM に接続：製品インスタンスがコントローラによって管理されている場合、コントローラはスケジュールされた時間に RUM レポートを送信します。

  Cisco DNA Center をコントローラとして使用している場合は、アドホックレポートのオプションがあります。必要なリリース（リリース 2.2.2 以降）の『Cisco DNA Center Administrator Guide』で「Manage Licenses」の「Upload Resource Utilization Details to CSSM」を参照してください。

- CSSM からの CSLU の切断：製品スタンスが CSLU に接続されている場合は、上記の「CSLU を介した CSSM への接続」に示したように製品インスタンスと同期してから、タスクCSSM へのエクスポート（CSLU インターフェイス）（215 ページ）、CSSM への使用状況データのアップロードと ACK のダウンロード（260 ページ）、CSSM からのインポート（CSLU インターフェイス）（216 ページ）を実行します。
- CSSM への接続なしで CSLU なし：特権 EXEC モードで **license smart save usage** コマンドを入力し、使用状況の必要な情報をファイルに保存します。次に、CSSMに接続しているワークステーションから、次のタスクを実行します。CSSMへの使用状況データのアップロードと ACK のダウンロード（260 ページ）> 製品インスタンスへのファイルのインストール（261 ページ）
- SSM オンプレミス展開：

  製品インスタンスを SSM オンプレミスと同期します。

  - 製品インスタンス開始型通信の場合：特権 EXEC モードで **license smart sync** コマンドを入力します。CSLU が現在 CSSM にログインしている場合、CSSM 内の関連付けられているスマートアカウントとバーチャルアカウントに自動的に送信されます。
  - SSM オンプレミス開始型通信の場合は、次の手順を実行します。SSM オンプレミス UI で、[Reports] > [Synchronization pull schedule] > [Synchronize now with the device] に移動します。

  使用状況情報を CSSM と同期します（いずれかを選択）。

  - SSM オンプレミスが CSSM に接続されている場合：SSM オンプレミス UI の [Smart Licensing] ワークスペースで、[Reports] > [Usage Schedules] > [Synchronize now with Cisco] に移動します。
  - SSM オンプレミスが CSSM に接続されていません。使用状況データのエクスポートとインポート（SSM オンプレミス UI）（235 ページ）を参照してください。

```
Error Message %SMART_LIC-6-TRUST_CODE_INSTALL_SUCCESS: A new licensing trust code
 was successfully installed on [chars].
```

説明：[chars] は、信頼コードが正常にインストールされた UDI です。

推奨するアクション：アクションは必要ありません。信頼コードがインストールされていることを確認するには、特権 EXEC モードで **show license status** コマンドを入力します。出力のヘッダー `Trust Code Installed:` で更新されたタイムスタンプを探します。

ポリシーを使用したスマートライセンシングのその他の参考資料

トピック	マニュアル タイトル
この章で使用するコマンドのシンタックスおよび使用方法の詳細については、必要なリリースのコマンドリファレンスで [System Mangement] > [System Mangement Commands] を参照してください。	Command Reference （Catalyst 9300 シリーズ スイッチ）
Cisco Smart Software Manager のヘルプ	Smart Software Manager Help
Cisco Smart License Utility（CSLU）のインストールおよびユーザーガイド	Cisco Smart License Utility クイック スタート セットアップ ガイド Cisco Smart License Utility ユーザーガイド

ポリシーを使用したスマートライセンシングの機能の履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

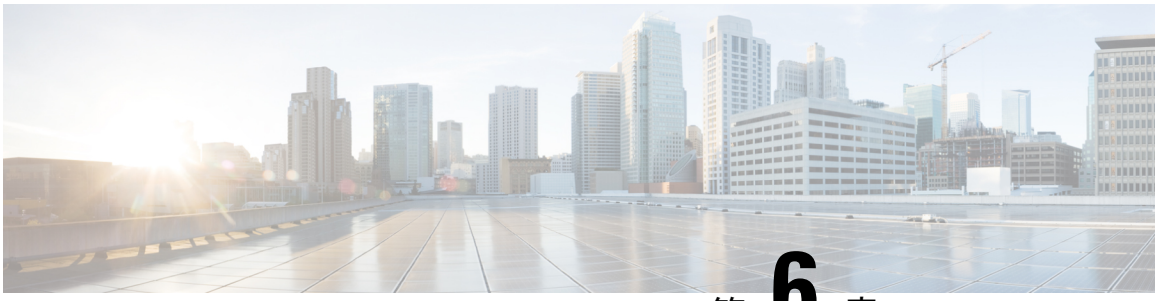
これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS XE Fuji 16.9.1	スマートライセンス	クラウドベースのソフトウェアライセンス管理ソリューションであり、ライセンス、ハードウェア、およびソフトウェアの使用状況の傾向を管理および追跡できます。 このリリース以降、スマートライセンスはデフォルトであり、ライセンスを管理するために使用できる唯一の方法です。 Cisco IOS XE Fuji 16.9.1 以降では、使用権（RTU）ライセンスモードが廃止され、関連する CLI の **license right-to-use** コマンドも使用できなくなりました。

リリース	機能	機能情報
Cisco IOS XE Amsterdam 17.3.2a	ポリシーを使用したスマートライセンス	スマートライセンシングの拡張バージョンには、ネットワークの運用を中断させないライセンスソリューションを提供するという主目的がありますが、むしろ、購入および使用しているハードウェアおよびソフトウェアライセンスを考慮して、コンプライアンス関係を実現するライセンスソリューションを提供するという目的もあります。 このリリース以降、ポリシーを使用したスマートライセンスがデバイスで自動的に有効になります。これは、このリリースにアップグレードする場合にも当てはまります。 デフォルトでは、CSSM のスマートアカウントとバーチャルアカウントは、ポリシーを使用したスマートライセンスで有効になっています。
	Cisco DNA Center での Smart Licensing Using Policy のサポート	Cisco DNA Center は、Cisco DNA Center リリース 2.2.2 以降、Smart Licensing Using Policy 機能をサポートしています。 Cisco DNA Center を使用して製品インスタンスを管理する場合、Cisco DNA Center は CSSM に接続し、CSSM とのすべての通信のインターフェイスとなります。 互換性のあるコントローラと製品インスタンスバージョンについては、コントローラ（141 ページ）を参照してください。 このトポロジについては、コントローラを介して CSSM に接続 （150 ページ）とトポロジのワークフロー：コントローラを介して CSSM に接続 （170 ページ）を参照してください。

リリース	機能	機能情報
Cisco IOS XE Amsterdam 17.3.3	Smart Licensing Using Policy 用の Smart Software Manager オンプレミス（SSM オンプレミス）サポート	SSM オンプレミスは、CSSM と連動するアセットマネージャです。これにより、CSSM に直接接続する代わりに、オンプレミスで製品とライセンスを管理できます。 互換性のある SSM オンプレミスと製品インスタンスバージョンについては、SSM オンプレミス （142 ページ）を参照してください。 このトポロジの概要についてと実装方法については、SSM オンプレミス展開 （153 ページ）とトポロジのワークフロー：SSM オンプレミス展開 （176 ページ）を参照してください。 既存のバージョンの SSM オンプレミスから、Smart Licensing Using Policy への移行をサポートするバージョンへの移行については、Smart Licensing Using Policy をサポートする SSM オンプレミスのバージョンへの移行 （208 ページ）を参照してください。
Cisco IOS XE Bengaluru 17.6.2	高セキュリティ（HSECK9）ライセンス	高セキュリティ（HSECK9）ライセンスは輸出規制対象のライセンスであり、Cisco Catalyst 9300X シリーズ スイッチで導入されました。サポートされている製品インスタンスで HSECK9 ライセンスの SLAC を取得してインストールできるようになりました。承認コード （143 ページ）を参照してください。 SLAC は次のトポロジで生成できます。 • トポロジのワークフロー：CSLU を介して CSSM に接続 （165 ページ） • トポロジのワークフロー：CSSM に直接接続 （168 ページ） • トポロジのワークフロー：CSLU は CSSM から切断 （171 ページ） • トポロジのワークフロー：CSSM への接続なし、CSLU なし （175 ページ） • トポロジのワークフロー：SSM オンプレミス展開 （176 ページ）

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



第 6 章

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

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

（注） この機能は、Cisco Catalyst 9500 シリーズ スイッチの C9500-32C、C9500-32QC、C9500-48Y4C、および C9500-24Y4C モデルではサポートされていません。

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、サーバー、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

ここでは、サポートされている AVC クラスマップとポリシーマップ形式について説明します。

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 *protocol name*	`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` ` police 5000000`	入力および出力

AVC ポリシーのフォーマット	AVC ポリシーの例	方向
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 police 4000000 class class-webex-category set dscp ef 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 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 client-up-child 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、VLANおよびその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。
- NBAR ベースの QoS ポリシー設定は、ポートチャネルメンバポートおよび SVI やサブ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ません。
- NBAR ベースの QoS ポリシー設定は、レイヤ 2 アクセスポートとトランクポート、およびレイヤ 3 ルーテッドポートでサポートされます。
- NBAR と送信（Tx）スイッチドポートアナライザ（SPAN）は、同じインターフェイスではサポートされません。
- プロトコルベースまたは属性ベースのいずれかのポートに同時に接続できるのは、NBAR ベースの QoS メカニズムの 1 つだけです。次の 2 つの属性のみがサポートされます。

- traffic-class
- business-relevance

- 従来の WDAVC QoS の制限事項は引き続き適用されます。
  - マーキングとポリシングのみがサポートされます。
  - 物理インターフェイスだけがサポートされます。
  - アプリケーション分類がオフラインで行われるため、QoS 分類には遅延があります（ただし、フローの最初のパケットは、正確な QoS 分類の前に転送されます）。
- 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）。
- 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ー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。
- 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。
- 接続は、物理レイヤ 2 およびレイヤ 3 ポートでのみ行う必要があります。これらのポートはポートチャネルの一部とすることはできません。トランクポートへの接続はサポートされません。
- パフォーマンス：各スイッチメンバは、50% 未満の CPU 使用率で、1 秒あたり 2000 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 20,000 の双方向フローと、24 個のアクセスポートごとに 10,000 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）。

- 有線 AVC では、この章の手順にリストされている固定のフィールドセットのみを使用できます。その他の組み合わせは使用できません。通常の FNF フローモニターでは、他の組み合わせも使用できます（サポートされている FNF フィールドのリストについては、『*Network Management Configuration Guide*』の「Configuring Flexible NetFlow」の章を参照してください）。
- Cisco IOS XE 16.12.1 リリース以降、新しいフローレコード（DNS フローレコード）が追加されました。DNS フローレコードは 5 タプルレコードに似ており、DNS ドメイン名フィールドが含まれています。DNS 関連のフィールドのみを考慮します。このレコードには、照合フィールドとしてのインターフェイスフィールドがないため、すべてのインターフェイスからの情報が同じレコードに集約されます。
- インターフェイスに AVC と ETA の両方が設定されている場合、インターフェイスに FNF を設定することはできません。
- IPv4 ユニキャストトラフィックに対してのみ、同じポートで AVC と ETA の両方を有効にできます。

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」のセクション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」のセクション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」のセクション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 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 レコードおよびフロー エクスポータに基づいて、フロー モニターを作成します。

- インターフェイスにフロー モニターを接続します。

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	**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`	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	**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。

	コマンドまたはアクション	目的
		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。 （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ないパケットはすべて **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：	分類したトラフィックにポリサーを定義します。

	コマンドまたはアクション	目的
	`Device(config-pmap-c)# police 100000 80000`	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 • *burst-byte* には、標準バースト サイズをバイト数で指定します。有効範囲は、1000 ～ 512000000 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。

	コマンドまたはアクション	目的
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 モニターの一致および収集フィールドのサポート強化

Cisco IOS XE Amsterdam 17.3.1 リリース以降、有線 AVC モニターで使用できる一致または収集フィールドに関する制限事項は削除されました。有線 AVC モニターでは、次に示す一致フィールドまたは収集フィールドの任意の組み合わせを使用できます。

サポートされる一致または収集フィールド

- application name
- ipv4/6 version
- ipv4/6 protocol
- transport tcp flags
- flow direction
- connection initiator
- connection client ipv4 address
- connection server ipv4 address
- ipv4 source address
- ipv4 destination address
- connection client ipv6 address
- connection server ipv6 address
- ipv6 source address
- ipv6 destination address
- connection client transport port
- connection server transport port
- transport source-port
- transport destination-port
- flow observation point
- interface input
- interface output
- datalink mac source address input

- datalink mac destination address input
- datalink mac source address output
- datalink mac destination address output
- datalink dot1q vlan input
- datalink dot1q vlan output

サポートされる収集専用フィールド

- connection client counter packets long
- connection client counter bytes network long
- connection server counter packets long
- connection server counter bytes network long
- counter bytes long
- counter packets long
- timestamp absolute first
- timestamp absolute last
- connection new-connections

抽出済みフィールド

- application dns domain-name
- application http host
- application ssl common-name

有線 AVC モニターでは、上記のフィールドを任意に組み合わせて設定できます。

フィールド「interface input」および「interface output」は、それぞれレコード内の一致フィールドとして使用できます。ただし、両方のフィールドを同じレコードの一致フィールドとして使用することはできません。両方のフィールドを収集フィールドとして使用することも、1 つを収集フィールドとして、1 つを同じレコード内の一致フィールドとして使用することもできます。

有線 AVC Flexible Netflow の設定

フロー レコードの作成

有線 AVC FNF は、従来の双方向フローレコードと方向性フローレコード（入力と出力）の 2 種類の定義済みフローレコードをサポートします。合計4つの異なる定義済みフローレコード（2 つの双方向フローレコードと 2 つの方向性フローレコード）を設定し、フローモニターに関連付けることができます。従来の双方向レコードはクライアント/サーバーアプリケーション統計情報レコードであり、新しい方向性レコードは入出力のアプリケーション統計情報です。

双方向フローレコード

フローレコード 1：双方向フローレコード

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 record** *flow_record_name* 例： `Device(config)# flow record fr-wdavc-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** *description* 例： `Device(config-flow-record)# description fr-wdavc-1`	（任意）フロー レコードの説明を作成します。
ステップ 4	**match ipv4 version** 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	**match ipv4 protocol** 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	**match application name** 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。
ステップ 8	**match connection server ipv4 address** 例： `Device(config-flow-record)# match connection server ipv4 address`	サーバー（フロー レスポンダ）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 9	**match connection server transport port** 例： `Device(config-flow-record)# match connection server transport port`	サーバー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。 • 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。

	コマンドまたはアクション	目的
ステップ 13	**collect connection new-connections** 例： `Device(config-flow-record)# collect connection new-connections`	観測された接続開始の数を収集するように指定します。
ステップ 14	**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 15	**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 16	**collect connection server counter packets long** 例： `Device(config-flow-record)# collect connection server counter packets long`	サーバー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 17	**collect connection server counter bytes network long** 例： `Device(config-flow-record)# collect connection server counter bytes network long`	サーバー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 18	**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 19	**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 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。

	コマンドまたはアクション	目的
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。

フローレコード 2：双方向フローレコード

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 record** *flow_record_name* 例： `Device(config)# flow record fr-wdavc-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** *description* 例： `Device(config-flow-record)# description fr-wdavc-1`	（任意）フローレコードの説明を作成します。
ステップ 4	**match ipv4 version** 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	**match ipv4 protocol** 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	**match application name** 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 client transport port** 例： `Device(config-flow-record)# match connection client transport port`	（任意）フローレコードのキーフィールドとして、クライアントの接続ポートとの一致を指定します。
ステップ 9	**match connection server ipv4 address** 例： `Device(config-flow-record)# match connection server ipv4 address`	サーバー（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 10	**match connection server transport port** 例： `Device(config-flow-record)# match connection server transport port`	サーバー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 11	**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 12	**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 13	**collect connection initiator** 例：	**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）

	コマンドまたはアクション	目的
	`Device(config-flow-record)# collect connection initiator`	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。 • 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 14	**collect connection new-connections** 例： `Device(config-flow-record)# collect connection new-connections`	観測された接続開始の数を収集するように指定します。
ステップ 15	**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 16	**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 17	**collect connection server counter packets long** 例： `Device(config-flow-record)# collect connection server counter packets long`	サーバー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 18	**collect connection server counter bytes network long** 例： `Device(config-flow-record)# collect connection server counter bytes network long`	サーバー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 19	**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。

	コマンドまたはアクション	目的
ステップ 20	**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 21	**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 22	**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。

方向性フローレコード

フローレコード 3：方向性フローレコード：入力

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 record** *flow_record_name* 例： `Device(config)# flow record fr-wdavc-3`	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** *description* 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	**match ipv4 version** 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	**match ipv4 protocol** 例： `Device(config-flow-record)# match ipv4 protocol`	IPv4 プロトコル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 6	**match ipv4 source address** 例： `Device(config-flow-record)# match ipv4 source address`	IPv4 送信元アドレスとの一致をキーフィールドとして指定します。
ステップ 7	**match ipv4 destination address** 例： `Device(config-flow-record)# match ipv4 destination address`	IPv4 宛先アドレスとの一致をキーフィールドとして指定します。
ステップ 8	**match transport source-port** 例： `Device(config-flow-record)# match transport source-port`	トランスポート発信元ポートとの一致をキーフィールドとして指定します。
ステップ 9	**match transport destination-port** 例： `Device(config-flow-record)# match transport destination-port`	トランスポート宛先ポートとの一致をキーフィールドとして指定します。
ステップ 10	**match interface input** 例： `Device(config-flow-record)# match interface input`	入力インターフェイスとの一致をキーフィールドとして指定します。
ステップ 11	**match application name** 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 12	**collect interface output** 例： `Device(config-flow-record)# collect interface output`	フローから出力インターフェイスを収集するように指定します。
ステップ 13	**collect counter bytes long** 例： `Device(config-flow-record)# collect counter bytes long`	フローのバイト数を収集するように指定します。
ステップ 14	**collect counter packets long** 例：	フローのパケット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 counter packets long`	
ステップ 15	**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 16	**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 17	**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 18	**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。

フローレコード 4：方向性フローレコード：出力

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 record** *flow_record_name* 例： `Device(config)# flow record fr-wdavc-4`	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** *description* 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	**match ipv4 version** 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。

	コマンドまたはアクション	目的
ステップ 5	**match ipv4 protocol** 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	**match ipv4 source address** 例： `Device(config-flow-record)# match ipv4 source address`	IPv4 送信元アドレスとの一致をキーフィールドとして指定します。
ステップ 7	**match ipv4 destination address** 例： `Device(config-flow-record)# match ipv4 destination address`	IPv4 宛先アドレスとの一致をキーフィールドとして指定します。
ステップ 8	**match transport source-port** 例： `Device(config-flow-record)# match transport source-port`	トランスポート発信元ポートとの一致をキーフィールドとして指定します。
ステップ 9	**match transport destination-port** 例： `Device(config-flow-record)# match transport destination-port`	トランスポート宛先ポートとの一致をキーフィールドとして指定します。
ステップ 10	**match interface output** 例： `Device(config-flow-record)# match interface output`	出力インターフェイスとの一致をキーフィールドとして指定します。
ステップ 11	**match application name** 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 12	**collect interface input** 例： `Device(config-flow-record)# collect interface input`	フローから入力インターフェイスを収集するように指定します。
ステップ 13	**collect counter bytes long** 例：	フローのバイト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 counter bytes long`	
ステップ 14	**collect counter packets long** 例： `Device(config-flow-record)# collect counter packets long`	フローのパケット数を収集するように指定します。
ステップ 15	**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 16	**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 17	**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 18	**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。

DNS フローレコード

フローレコード 5：DNS フローレコード

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 record** *flow_record_name* 例： `Device(config)# flow record fr-wdavc-5`	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** *description* 例：	（任意）フロー レコードの説明を作成します。

	コマンドまたはアクション	目的
	`Device(config-flow-record)# description flow-record-5`	
ステップ 4	**match ipv4 version** 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	**match ipv4 protocol** 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	**match application name** 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。
ステップ 8	**match connection client transport port** 例： `Device(config-flow-record)# match connection client transport port`	フローレコードのキーフィールドとして、クライアントの接続ポートとの一致を指定します。
ステップ 9	**match connection server ipv4 address** 例： `Device(config-flow-record)# match connection server ipv4 address`	サーバー（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 10	**match connection server transport port** 例： `Device(config-flow-record)# match connection server transport port`	サーバー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 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

	コマンドまたはアクション	目的
		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 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 13	**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 14	**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。 • 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 15	**collect connection new-connections** 例： `Device(config-flow-record)# collect connection new-connections`	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
ステップ 16	**collect connection server counter packets long** 例： `Device(config-flow-record)# collect connection server counter packets long`	サーバー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 17	**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 18	**collect connection server counter bytes network long** 例： `Device(config-flow-record)# collect connection server counter bytes network long`	サーバー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 19	**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 20	**collect application dns domain-name** 例： `Device(config-flow-record)# collect application dns domain-name`	DNS ドメイン名を DNS フローレコードの収集フィールドとして使用するように設定します。
ステップ 21	**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。

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	**flow exporter** *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** *description* 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** *seconds*] 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。
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 モニターの作成

フロー モニターを作成して、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`	フローモニターを作成し、フローモニター 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** *description* 例： `Device(config-flow-monitor)# description flow-monitor-1`	（任意）フローモニター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。
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。

	コマンドまたはアクション	目的
ステップ 8	**show flow monitor** 例： `Device# show flow monitor`	すべてのフローモニターに関する情報を表示します。
ステップ 9	**show flow monitor** *flow-monitor-name* 例： `Device# show flow monitor flow-monitor-1`	指定した有線 AVC フロー モニターに関する情報を表示します。
ステップ 10	**show flow monitor** *flow-monitor-name* **statistics** 例： `Device# show flow monitor flow-monitor-1 statistics`	有線 AVC フロー モニターの統計情報を表示します。
ステップ 11	**clear flow monitor** *flow-monitor-name* **statistics** 例： `Device# clear flow monitor flow-monitor-1 statistics`	指定したフローモニターの統計情報をクリアします。**clear flow monitor flow-monitor-1 statistics** を使用した後に **show flow monitor flow-monitor-1 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たことを確認します。
ステップ 12	**show flow monitor** *flow-monitor-name* **cache format table** 例： `Device# show flow monitor flow-monitor-1 cache format table`	表形式でフローキャッシュの内容を表示します。
ステップ 13	**show flow monitor** *flow-monitor-name* **cache format record** 例： `Device# show flow monitor flow-monitor-1 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** *flow-monitor-name* **cache format csv** 例： `Device# show flow monitor flow-monitor-1 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 モニターの関連付け

異なる事前定義済みレコードを持つ 2 つの異なる有線 AVC モニターをインターフェイスに同時に接続できます。

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	**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 モニターを関連付けます。
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、ユーザーは、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
- 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ーのホスト名
- **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 サーバー名指定（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ーリング

カスタム アプリケーションのモニターリングのための 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*

```
Device# 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。

Warning スイッチスタック構成を使用する場合は、各スイッチに同じプロトコルパックファイルがロードされていることを確認します。スタック内のプライマリスイッチで **ip nbar protocol-pack flash** *protocol-pack-file* コマンドを実行すると、ファイルがロードされていないスタック内のスイッチは、設定の不一致が原因でリロードされます。

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/home

NBAR2 プロトコル パックの前提条件

新しいプロトコル パックをロードする前に、すべてのスイッチ メンバー上でプロトコル パックをフラッシュにコピーする必要があります。

プロトコル パックをロードするには、NBAR2 プロトコル パックのロード（321 ページ） を参照してください。

NBAR2 プロトコル パックのロード

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 モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル設定モードを開始します。
ステップ 3	**ip nbar protocol-pack** *protocol-pack* [**force**] 例： `Device(config)# ip nbar protocol-pack flash:defProtoPack` 例： `Device(config)# default ip nbar protocol-pack`	プロトコル パックをロードします。 • 基本のプロトコルパックバージョンとは異なる、より低いバージョンのプロトコルパックを指定し、ロードするには、**force** キーワードを使用します。これにより、スイッチの現在のプロトコル パックでサポートされていない設定も削除されます。 組み込みのプロトコル パックに戻るには、次のコマンドを使用します。

	コマンドまたはアクション	目的
ステップ 4	**exit** 例： `Device(config)# exit`	特権 EXEC モードに戻ります。
ステップ 5	**show ip nbar protocol-pack** {protocol-pack | **active**} [**detail**] 例： `Device# show ip nbar protocol-pack active`	プロトコル パック情報を表示します。 • このコマンドを使用して、ロードされたプロトコル パックのバージョン、パブリッシャ、その他の詳細を確認します。 • 指定されたプロトコル パックの情報を表示するには、*protocol-pack* 引数を使用します。 • アクティブなプロトコルパックの情報を表示するには、**active** キーワードを使用します。 • 詳細なプロトコルパックの情報を表示するには、**detail** キーワードを使用します。

例：*NBAR2* プロトコル パックのロード

次の例に、新しいプロトコル パックをロードする方法を示します。

```
Device> enable
Device# configure terminal
Device(config)# ip nbar protocol-pack flash:newDefProtoPack
Device(config)# exit
```

次の例に、**force** キーワードを使用して下位バージョンのプロトコルパックをロードする方法を示します。

```
Device> enable
Device# configure terminal
Device(config)# ip nbar protocol-pack flash:OldDefProtoPack force
Device(config)# exit
```

次の例に、組み込みのプロトコル パックに戻す方法を示します。

```
Device> enable
Device# configure terminal
Device(config)# default ip nbar protocol-pack
Device(config)# exit
```

Application Visibility and Control のモニターリング

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、スイッチ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ーするために使用できます。

表 *20*：スイッチのアプリケーションの可視性モニターリングコマンド

コマンド	目的
show ip nbar protocol-discovery [**interface** *interface-type interface-number*] [**stats**{**byte-count** | **bit-rate** | **packet-count** | **max-bit-rate**}] [**protocol** *protocol-name* | **top-n** *number*]	NBAR Protocol Discovery 機能によって収集された統計情報を表示します。 • （任意）表示される統計情報を最適化するには、キーワードおよび引数を入力します。キーワードのそれぞれの詳細については、『Cisco IOS Quality of Service Solutions Command Reference』の**show ip nbar protocol-discovery** コマンドを参照してください。
show policy-map interface *interface-type interface-number*	インターフェイスに適用したポリシーマップについての情報を表示します。
show platform software fed switch スイッチ *ID* **wdavc flows**	指定したスイッチのすべてのフローに関する統計情報を表示します。

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
```

次に、NBAR 属性に基づい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-all rel-relevant
Device(config-cmap)# match protocol attribute business-relevance business-relevant

Device(config)# class-map match-all rel-irrelevant
Device(config-cmap)# match protocol attribute business-relevance business-irrelevant

Device(config)# class-map match-all rel-default
Device(config-cmap)# match protocol attribute business-relevance default

Device(config)# class-map match-all class--ops-admin-and-rel
Device(config-cmap)# match protocol attribute traffic-class ops-admin-mgmt
Device(config-cmap)# match protocol attribute business-relevance business-relevant
```

次に、NBAR 属性に基づくクラスマップに基づいてポリシーマップを作成する例を示します。

```
Device# configure terminal
Device(config)# policy-map attrib--rel-types
Device(config-pmap)# class rel-relevant
Device(config-pmap-c)# set dscp ef
Device(config-pmap-c)# class rel-irrelevant
Device(config-pmap-c)# set dscp af11
Device(config-pmap-c)# class rel-default
Device(config-pmap-c)# set dscp default

Device(config)# policy-map attrib--ops-admin-and-rel
Device(config-pmap)# class class--ops-admin-and-rel
Device(config-pmap-c)# set dscp cs5
```

次に、NBAR 属性に基づくポリシーマップを有線ポートに適用する例を示します。

```
Device# configure terminal
Device(config)# interface GigabitEthernet1/0/2
Device(config-if)# service-policy input attrib--rel-types
```

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。

```
Device# 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。

```
Device# 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 コマンドによる属性ベースの QoS 設定の表示

show policy-map interface

すべてのインターフェイス上の属性ベースの 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。

```
Device# show policy-map interface gigabitEthernet 1/0/2
 GigabitEthernet1/0/2

  Service-policy input: attrib--rel-types

    Class-map: rel-relevant (match-all)
      20 packets
      Match: protocol attribute business-relevance business-relevant
      QoS Set
        dscp ef

    Class-map: rel-irrelevant (match-all)
      0 packets
      Match: protocol attribute business-relevance business-irrelevant

      QoS Set
        dscp af11

    Class-map: rel-default (match-all)
      14 packets
      Match: protocol attribute business-relevance default
      QoS Set
        dscp default

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

show ip nbar protocol-attribute

NBAR で使用されるすべてのプロトコル属性を表示します。

次に、一部の属性の出力例を示します。

```
Device# show ip nbar protocol-attribute cisco-jabber-im
           Protocol Name : cisco-jabber-im
                encrypted : encrypted-yes
                   tunnel : tunnel-no
                 category : voice-and-video
             sub-category : enterprise-media-conferencing
       application-group : cisco-jabber-group
          p2p-technology : p2p-tech-no
           traffic-class : transactional-data
      business-relevance : business-relevant
         application-set : collaboration-apps

Device# show ip nbar protocol-attribute google-services
            Protocol Name : google-services
                encrypted : encrypted-yes
                   tunnel : tunnel-no
                 category : other
              sub-category : other
       application-group : google-group
          p2p-technology : p2p-tech-yes
           traffic-class : transactional-data
      business-relevance : default
         application-set : general-browsing

Device# show ip nbar protocol-attribute dns
            Protocol Name : google-services
                encrypted : encrypted-yes
                   tunnel : tunnel-no
                 category : other
              sub-category : other
       application-group : google-group
          p2p-technology : p2p-tech-yes
           traffic-class : transactional-data
      business-relevance : default
         application-set : general-browsing

Device# show ip nbar protocol-attribute unknown
            Protocol Name : unknown
                encrypted : encrypted-no
                   tunnel : tunnel-no
                 category : other
              sub-category : other
       application-group : other
          p2p-technology : p2p-tech-no
           traffic-class : bulk-data
      business-relevance : default
         application-set : general-misc
```

show コマンドによるフロー モニター設定の表示

show flow monitor wdavc

指定した有線 AVC フロー モニター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 モニター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 モニターの統計情報をクリアします。**clear flow monitor wdavc statistics** を使用した後に **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たことを確認します。以下に、フローモニター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。どうしてですか。

    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	マニュアル タイトル
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 9300* シリーズ スイッチ）

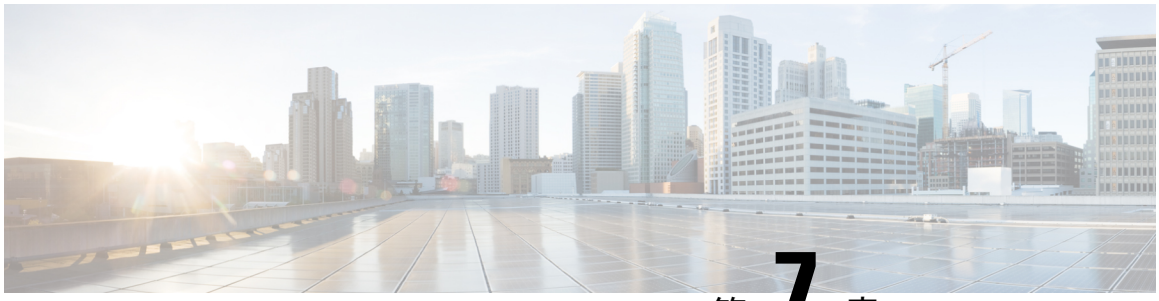
有線ネットワークでの Application Visibility and Control の機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

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

リリース	機能	機能情報
Cisco IOS XE Everest 16.5.1a	有線ネットワークでの 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。
Cisco IOS XE Fuji 16.8.1a	有線アプリケーションの表示およびコントロール（有線 AVC）属性ベース QoS（EasyQoS）	特定のプロトコルではなく、Network-Based Application Recognition（NBAR）属性に基づいて QoS クラスとポリシーを定義できるようになりましたが、いくつかの制限があります。サポートされる NBAR 属性は、business-relevance および traffic-class のみです。
Cisco IOS XE Gibraltar 16.12.1	DNS フローレコード	DNS フローレコードのサポートが導入されました。DNS フローレコードは、フローレコードを定義するための collect フィールドとして DNS ドメイン名を使用します。
Cisco IOS XE Amsterdam 17.3.1	アプリケーションの可視性およびコントロールと暗号化トラフィック分析の相互運用性	同じポートでのアプリケーションの表示およびコントロールと暗号化トラフィック分析の相互運用性のサポートが導入されました。

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



第 7 章

SDM テンプレートの設定

SDM テンプレートに関する情報

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。

Cisco Catalyst 9300 シリーズ スイッチは、次のテンプレートをサポートしています。

- アクセス
- NAT

SDM テンプレートに変更を加えたらすぐにシステムをリロードすることを推奨します。テンプレートを変更し、システムを再起動した後、 **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 テンプレートの設定方法

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 モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル設定モードを開始します。
ステップ 3	**sdm prefer access | nat** 例： `Device(config)# sdm prefer access`	スイッチをアクセステンプレートに設定します。
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
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ーリングおよびメンテナンス

SDM テンプレートの確認

SDM テンプレートをモニターおよびメンテナンスするには、次のコマンドを使用します。

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。

（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、該当する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 テンプレートを削除すると、そのような他の関連するコマンドも削除されるため、明示的に再設定しなければならなくなります。

SDM テンプレートの設定例

例：SDM テンプレートの表示

次に、Cisco Catalyst 9300 シリーズ スイッチのアクセステンプレート情報を表示する出力例を示します。

```
Device# show sdm prefer access
This is the Access template.
  Number of VLANs:                                4094
  Unicast MAC addresses:                          32768
  Overflow Unicast MAC addresses:                 1024
  L2 Multicast entries:                           8192
  Overflow L2 Multicast entries:                  512
  L3 Multicast entries:                           8192
  Overflow L3 Multicast entries:                  512
  Directly connected routes:                      24576
  Indirect routes:                                8192
  STP Instances:                                  1024
  Security Access Control Entries:                5120
  QoS Access Control Entries:                     5120
  Policy Based Routing ACEs:                      1024
  Netflow Input ACEs:                             256
  Netflow Output ACEs:                            768
  Ingress Netflow ACEs:                           256
  Egress Netflow ACEs:                            768
  Flow SPAN ACEs:                                 1024
  Tunnels:                                        512
  LISP Instance Mapping Entries:                  512
  Control Plane Entries:                          512
  Input Netflow flows:                            32768
  Output Netflow flows:                           32768
  SGT/DGT (or) MPLS VPN entries:                  8192
  SGT/DGT (or) MPLS VPN Overflow entries:         512
  Wired clients:                                  2048
```

```
  MACSec SPD Entries:                               256
  MPLS L3 VPN VRF:                                  255
  MPLS Labels:                                      2048
  MPLS L3 VPN Routes VRF Mode:                      7168
  MPLS L3 VPN Routes Prefix Mode:                   3072
  MVPN MDT Tunnels:                                 256
  L2 VPN EOMPLS Attachment Circuit:                 256
  MAX VPLS Bridge Domains :                         128
  MAX VPLS Peers Per Bridge Domain:                 32
  MAX VPLS/VPWS Pseudowires :                       1024
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.
* values can be modified by sdm cli.
```

次に、Cisco Catalyst 9300 シリーズ スイッチの NAT テンプレート情報を表示する出力例を示します。

```
Device# show sdm prefer nat
This is the NAT template.
  Number of VLANs:                                  4094
  Unicast MAC addresses:                            32768
  Overflow Unicast MAC addresses:                   1024
  L2 Multicast entries:                             8192
  Overflow L2 Multicast entries:                    512
  L3 Multicast entries:                             8192
  Overflow L3 Multicast entries:                    512
  Directly connected routes:                        24576
  Indirect routes:                                  8192
  Security Access Control Entries:                  5120
  QoS Access Control Entries:                       1024
  Policy Based Routing ACEs:                        5120
  Netflow Input ACEs:                               256
  Netflow Output ACEs:                              768
  Flow SPAN ACEs:                                   1024
  Tunnels:                                          512
  LISP Instance Mapping Entries:                    512
  Control Plane Entries:                            512
  Input Netflow flows:                              32768
  Output Netflow flows:                             32768
  SGT/DGT (or) MPLS VPN entries:                    8192
  SGT/DGT (or) MPLS VPN Overflow entries:           512
  Wired clients:                                    2048
  MACSec SPD Entries:                               256
  MPLS L3 VPN VRF:                                  255
  MPLS Labels:                                      2048
  MPLS L3 VPN Routes VRF Mode:                      7168
  MPLS L3 VPN Routes Prefix Mode:                   8192
  MVPN MDT Tunnels:                                 256
  L2 VPN EOMPLS Attachment Circuit:                 256
  MAX VPLS Bridge Domains :                         128
  MAX VPLS Peers Per Bridge Domain:                 32
  MAX VPLS/VPWS Pseudowires :                       1024
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.
* values can be modified by sdm cli.
```

次の出力例は、Cisco Catalyst 9300 シリーズ スイッチの Cisco IOS XE Amsterdam 17.3.1 以降のリリースの C9300-24UB、C9300-24UXB、および C9300-48UB モデルでのアクセステンプレート情報を示しています。

```
Device# show sdm prefer access
 Number of VLANs:                                  4094
```

```
  Unicast MAC addresses:                          49152
  Overflow Unicast MAC addresses:                 1024
  L2 Multicast entries:                           16384
  Overflow L2 Multicast entries:                  1024
  L3 Multicast entries:                           32768
  Overflow L3 Multicast entries:                  1024
  Directly connected routes:                      49152
  Indirect routes:                                65536
  Security Access Control Entries:                18432
  QoS Access Control Entries:                     6144
  Policy Based Routing ACEs / NAT ACEs:           14336
  Netflow Input ACEs:                             1024
  Netflow Output ACEs:                            2048
  Flow SPAN ACEs:                                 1024
  Tunnels:                                        1024
  LISP Instance Mapping Entries:                  2048
  Control Plane Entries:                          512
  Input Netflow flows:                            65536
  Output Netflow flows:                           65536
  SGT/DGT (or) MPLS VPN entries:                  8192
  SGT/DGT (or) MPLS VPN Overflow entries:         512
  Wired clients:                                  2048
  MACSec SPD Entries:                             1024
  VRF:                                            256
  MPLS Labels:                                    12288
  MPLS L3 VPN Routes VRF Mode:                    32768
  MPLS L3 VPN Routes Prefix Mode:                 8192
  MVPN MDT Tunnels:                               1024
  L2 VPN EOMPLS Attachment Circuit:               1024
  MAX VPLS Bridge Domains :                       128
  MAX VPLS Peers Per Bridge Domain:               32
  MAX VPLS/VPWS Pseudowires :                     4096

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 access
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	マニュアル タイトル
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 9300* シリーズ スイッチ）

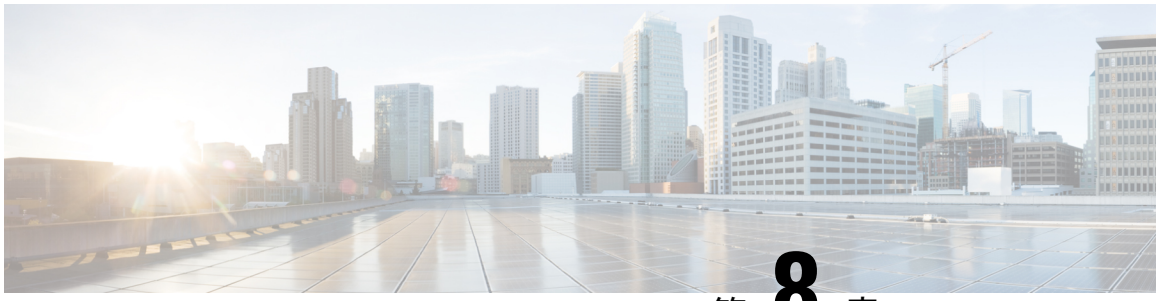
SDM テンプレートの機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

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

リリース	機能	機能情報
Cisco IOS XE Everest 16.5.1a	SDM テンプレート	標準の SDM テンプレートを使用すると、システムリソースを設定して、特定の機能のサポートを最適化できます。
Cisco IOS XE Gibraltar 16.12.3	C9300-24UB、C9300-24UXB、および C9300-48UB デバイスのスケーラビリティメトリックの変更	Cisco Catalyst 9300 シリーズ スイッチの C9300-24UB、C9300-24UXB、および C9300-48UB モデルで、次の機能の転送スケール番号が変更されました • レイヤ 2 ユニキャスト MAC アドレス：49152 • レイヤ 3 マルチキャスト：32768 • QoS アクセス コントロール エントリ：6144 • ポリシーベースルーティング ACE/NAT ACE：14336
Cisco IOS XE Amsterdam 17.3.1	C9300-24UB、C9300-24UXB、および C9300-48UB デバイスのスケーラビリティメトリックの変更	Cisco Catalyst 9300 シリーズ スイッチの C9300-24UB、C9300-24UXB、および C9300-48UB モデルで、次の機能の転送スケール番号が変更されました • レイヤ 2 ユニキャスト MAC アドレス：49152 • レイヤ 3 マルチキャスト：32768 • QoS アクセス コントロール エントリ：6144 • ポリシーベースルーティング ACE/NAT ACE：14336

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



第 8 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。スタック内のメンバスイッチはシステムメッセージをトリガーできます。システムメッセージを生成するメンバスイッチは、ホスト名を hostname-n の形式（n はスイッチ）で付加し、出力をアクティブスイッチのロギングプロセスにリダイレクトします。アクティブスイッチはスタックメンバですが、そのホスト名はシステムメッセージの末尾に追加されません。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 サーバーなど）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 サーバー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。スイッチスタックの場合は、アクティブスイッチ上に保存します。スタンドアロンスイッチまたはアクティブスイッチに障害が発生すると、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 サーバー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。スイッチスタックでは、すべてのメンバスイッチコンソールにより、同じコンソール出力が用意されます。

（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 *21* : 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にのみ、ログメッセージにシーケンス番号をスタンプします。

要素	説明
timestamp formats: *mm/dd h h: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。この情報が表示されるのは、**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のみです。
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。
MNEMONIC	メッセージを一意に示すテキスト ストリングです。
description	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。
hostname-n（ホスト名 *-n*）	スタック メンバーのホスト名およびスタック内のスイッチ番号。アクティブスイッチはスタックメンバですが、そのホスト名はシステムメッセージの末尾に追加されません。

デフォルトのシステム メッセージ ロギングの設定

表 *22*：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	イネーブル
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 サーバー	ディセーブル

機能	デフォルト設定
Syslog サーバーの IP アドレス	未設定
サーバー機能	local7
サーバーの重大度	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に送信されるようにsyslogメッセージトラップが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に達している場合）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。

	コマンドまたはアクション	目的
ステップ 2	**logging buffered** [*size*] 例： `Device(config)# logging buffered 8192`	スイッチ上、ログメッセージを内部バッファに保存します。. 指定できる範囲は 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。 （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** *host* 例： `Device(config)# logging 125.1.1.100`	UNIX Syslog サーバー ホストにメッセージを保存します。 *host* には、syslog サーバー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 サーバー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** — には、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 ～

	コマンドまたはアクション	目的
		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。
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	**terminal monitor** 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、ユーザー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ー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。
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

	コマンドまたはアクション	目的
		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。
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。 • **service timestamps log uptime** • **service timestamps log datetime**[**msec** | **localtime** | **show-timezone**] 例： `Device(config)# service timestamps log uptime`	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

	コマンドまたはアクション	目的
	または `Device(config)# service timestamps log datetime`	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。
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 サーバーに保存するメッセージを制限します。 デフォルトで、Syslog サーバーは通知メッセージ、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。
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 サーバーに送信される 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** メッセージは送信されません。
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 サーバーに送信する前に、UNIX サーバー上で 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。 例： `local7.debug /usr/adm/logs/cisco.log`	• **local7**：ロギング機能を指定します。 • **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ー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 ログのモニターリング

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：システム メッセージのスタック構成

次の例では、アクティブスイッチの部分的なスイッチシステムメッセージとスタックメンバ（ホスト名は *Switch-2*）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 GigabitEthernet1/0/1, changed state to up
00:00:47: %LINK-3-UPDOWN: Interface GigabitEthernet1/0/2, 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 GigabitEthernet1/0/1, 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)


00:00:46: %LINK-3-UPDOWN: Interface Port-channel1, changed state to up (Switch-2)
00:00:47: %LINK-3-UPDOWN: Interface GigabitEthernet2/0/1, changed state to up (Switch-2)
00:00:47: %LINK-3-UPDOWN: Interface GigabitEthernet2/0/2, changed state to up (Switch-2)
00:00:48: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to down
(Switch-2)
00:00:48: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet2/0/1, changed
 state to down 2 (Switch-2)
```

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	マニュアル タイトル
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 9300* シリーズ スイッチ）

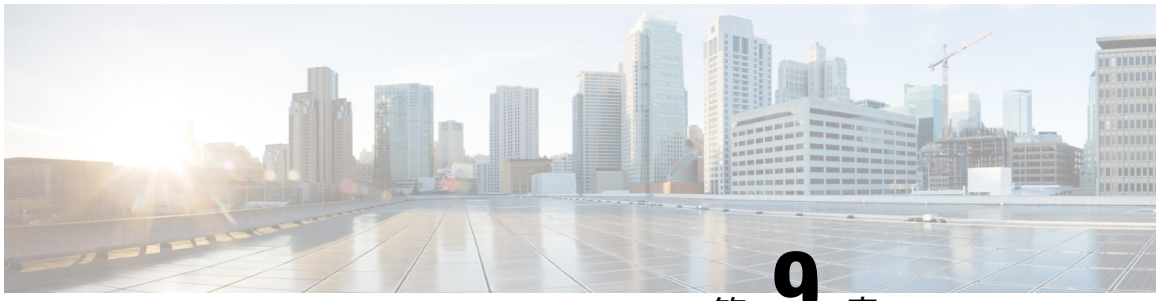
システムメッセージログの機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

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

リリース	機能	機能情報
Cisco IOS XE Everest 16.5.1a	システムメッセージログ	システムメッセージ出力は、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 サーバーなど）に配信する処理を制御します

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



第 9 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

オンライン診断機能を使用すると、デバイスをアクティブネットワークに接続したまま、デバイスのハードウェア機能をテストして確認できます。オンライン診断には、個別のハードウェアコンポーネントを確認して、データパスおよび制御信号を検証するパケットスイッチングテストが含まれます。

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、ヘルスモニター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、ユーザーが指定した間隔または指定した時刻に実行されます。ヘルスモニターリングは、バックグラウンドでユーザーが指定した間隔で実行されます。ヘルスモニターリングテストは、テストに基づいて 90、100、または 150 秒ごとに実行されます。

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。

Generic Online Diagnostics（GOLD）テスト

（注）
- オンライン診断テストをイネーブルにする前に、コンソールロギングをイネーブルにしてすべての警告メッセージを表示してください。
- テストの実行中、ポートを内部的にループしてストレステストを行いますが、外部トラフィックがテスト結果に影響を与えることがあるため、すべてのポートがシャットダウンされます。スイッチを正常な稼働に戻すために、スイッチをリロードします。スイッチをリロードするコマンドを実行すると、コンフィギュレーションを保存するかどうかを尋ねられます。コンフィギュレーションは保存しないでください。
- 他のモジュール上でテストを実行している場合、テストが開始され、完了したら、モジュールをリセットする必要があります。

ここでは、GOLD テストについて説明します。

DiagGoldPktTest

この GOLD パケットループバックテストは、MAC レベルのループバック機能を検証します。このテストでは、ハードウェアで Unified Access Data Plane（UADP; ユニファイドアクセスデータ プレーン）ASIC によってサポートされる GOLD パケットが送信されます。このパケットは MAC レベルでループバックし、保存されているパケットと照合されます。

属性	説明
ディスラプティブまたはノンディスラプティブ	ノンディスラプティブ
推奨事項	要件に従ってこのオンデマンドテストを実行します。
デフォルト	オフ
最初のリリース	Cisco IOS XE Everest 16.6.1
修正処置	–
ハードウェア サポート	スーパバイザ

DiagThermalTest

このテストは、デバイスセンサーからの温度の読み取り値を検証します。

属性	説明
ディスラプティブまたはノンディスラプティブ	ノンディスラプティブ

属性	説明
推奨事項	ディセーブルにしないでください。これはオンデマンドテストとして実行し、管理者がダウン状態の場合はヘルスモニターリングテストとして実行します。
デフォルト	オン
最初のリリース	Cisco IOS XE Everest 16.6.1
修正処置	–
ハードウェア サポート	スーパバイザ

DiagFanTest

このテストは、すべてのファンモジュールが挿入され、ボード上で正しく動作していることを検証します。

属性	説明
ディスラプティブまたはノンディスラプティブ	ノンディスラプティブ
推奨事項	ファンモジュールに問題が発生した場合は、ヘルスモニターリングテストとしてこれを実行します。
デフォルト	オン
最初のリリース	Cisco IOS XE Everest 16.6.1
修正処置	–
ハードウェア サポート	スーパバイザ

DiagPhyLoopbackTest

この PHY ループバックテストは、PHY レベルのループバック機能を検証します。このテストでは、PHY レベルでループバックし、保存されているパケットと照合されるパケットが送信されます。ヘルスモニターリングテストとして実行することはできません。

（注） このテストがオンデマンドで実行される特定のケースでは、ポートは error-disabled ステートに移行します。このような場合は、インターフェイス コンフィギュレーション モードで **shut** および **no shut** コマンドを使用して、これらのポートを再度イネーブルにします。

属性	説明
ディスラプティブまたはノンディスラプティブ	ディスラプティブ
推奨事項	外部コネクタへのリンクがダウンしている場合は、このオンデマンドテストを実行してリンクの正常性を確認します。
デフォルト	オフ
最初のリリース	Cisco IOS XE Everest 16.6.1
修正処置	–
ハードウェア サポート	スーパバイザ

DiagScratchRegisterTest

このスクラッチ登録テストは、レジスタに値を書き込み、これらのレジスタからその値を読み取ることで、ASIC の正常性をモニターします。

属性	説明
ディスラプティブまたはノンディスラプティブ	ノンディスラプティブ
推奨事項	ディセーブルにしないでください。このテストは、レジスタに値を書き込むタスクが失敗した場合に実行します。これは、ヘルスモニターリングテストとしても、オンデマンドテストとしても実行できます。
デフォルト	オン
最初のリリース	Cisco IOS XE Everest 16.6.1
修正処置	–
ハードウェア サポート	スーパバイザ

DiagPoETest

このテストは、PoEコントローラ機能をチェックします。通常のスイッチ動作中は、このテストを実行しないでください。

属性	説明
ディスラプティブまたはノンディスラプティブ	ノンディスラプティブ

属性	説明
推奨事項	このテストは、ポートで PoE コントローラの問題が発生した場合に実行します。これは、オンデマンドテストとしてのみ実行できます。
デフォルト	オフ
最初のリリース	Cisco IOS XE Everest 16.6.1
修正処置	–
ハードウェア サポート	ラインカード

DiagStackCableTest

このテストは、スタック構成環境のスタックリングループバック機能を検証します。ヘルスモニターリングテストとして実行することはできません。

属性	説明
ディスラプティブまたはノンディスラプティブ	ディスラプティブ
推奨事項	このテストを実行し、スタック構成環境のスタックリングループバック機能を検証します。
デフォルト	オフ
最初のリリース	Cisco IOS XE Everest 16.6.1
修正処置	テストに失敗した場合は、スタックケーブルとコネクタを確認してください。
ハードウェア サポート	スーパバイザ

DiagMemoryTest

この詳細な ASIC メモリテストは、通常のスイッチ動作中に実行します。このテストでは、スイッチはメモリの組み込み自己診断テストを使用します。メモリテストでは、テスト後にスイッチを再起動する必要があります。

属性	説明
ディスラプティブまたはノンディスラプティブ	非常にディスラプティブです。
推奨事項	このオンデマンドテストは、システムでメモリ関連の問題が発生した場合にのみ実行します。テスト対象のスーパバイザエンジンをリ

属性	説明
	ロードしない場合は、このテストを実行しないでください。
デフォルト	オフ
最初のリリース	Cisco IOS XE Everest 16.6.1
修正処置	–
ハードウェア サポート	スーパバイザ

TestUnusedPortLoopback

このテストは、実行時にスーパバイザモジュールとモジュールのネットワークポート間のデータパスを定期的に確認し、着信ネットワーク インターフェイス ポートがロックされているかどうかを判断します。このテストでは、レイヤ2のパケットはテストポートおよびスーパバイザエンジンのインバンドポートに関連付けられた VLAN にフラッディングされます。パケットはテストポート内をループバックして、同じ VLAN のスーパバイザエンジンに戻ります。このテストは、ケーブルが接続されているかどうかに関係なく、未使用の（管理上のダウン、つまりポートがシャットダウンされている）ネットワークポートでのみ実行され、ポートあたり 1 ミリ秒以内に完了します。このテストは、現在の ASIC にノンディスラプティブ ループバック テストがないため、代用として使用され、60 秒ごとに実行されます。

属性	説明
ディスラプティブまたはノンディスラプティブ	ノンディスラプティブ
推奨事項	ディセーブルにしないでください。CPU 使用率の急上昇中、このテストは精度を維持するために自動的にディセーブルになります。
デフォルト	オン
最初のリリース	Cisco IOS XE Fuji 16.9.1
修正処置	ポートに障害が発生したことを示すsyslogメッセージを表示します。スーパバイザエンジン以外のモジュールでは、すべてのポートグループに障害が発生した場合（たとえば、ポート ASIC ごとに最低 1 つのポートで、すべてのポート ASICの障害しきい値より多く障害が発生した場合）、デフォルトのアクションではモジュールがリセットされ、リセットを 2 回行ったあとにモジュールの電源を切断します。
ハードウェア サポート	スーパバイザ

TestPortTxMonitoring

このテストは、ステータスが UP のデバイスに物理的に接続されている各ネットワークポートの送信方向のデータパストラフィックを定期的にモニターします。このテストは、ポートあたり 1 ミリ秒以内に完了します。また、このテストでは、ASIC レベルで送信カウンタをモニターして、ポートがスタックしていないことを確認します。テストではsyslogメッセージが表示され、ユーザーは Cisco IOS Embedded Event Manager（EEM）を使用して修正アクションを実行できます。

diagnostic monitor interval および **diagnostic monitor threshold** コマンドをそれぞれ入力して、時間間隔としきい値を設定します。テストでは、パケットを送信する Cisco Discovery Protocol を利用します。テストは 75 秒ごとに実行され、障害しきい値はデフォルトで 5 秒に設定されています。

属性	説明
ディスラプティブまたはノンディスラプティブ	ノンディスラプティブ
推奨事項	ディセーブルにしないでください。
デフォルト	オン
最初のリリース	Cisco IOS XE Everest 16.9.1
修正処置	ポートに障害が発生したことを示すsyslogメッセージを表示します。
ハードウェア サポート	スーパバイザ

オンライン診断の設定方法

ここでは、オンライン診断設定を構成するさまざまな手順について説明します。

オンライン診断テストの開始

デバイスで実行する診断テストを設定したあと、**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、テストプロセスの途中停止はできません。

手動でオンライン診断テストを開始するには、**diagnostic start switch** 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

診断モニター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、**diagnostic schedule switch** 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	**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	**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。 • **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。

	コマンドまたはアクション	目的
		• 毎週：**weekly** *day-of-week hh:mm* パラメータを使用します。

ヘルス モニターリング診断の設定

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ーリング診断テストを設定できます。各ヘルスモニターリングテストの実行間隔を設定したり、デバイスをイネーブルにし、テスト失敗時の Syslog メッセージを生成したり、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、ヘルスモニターリングはいくつかのテストでのみイネーブルであり、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。

ヘルスモニター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 モードを有効にします。 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	**diagnostic monitor interval switch** *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、ヘルスモニター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。

	コマンドまたはアクション	目的
		• *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*：モニターリング間隔（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*：モニターリング間隔（ミリ秒（ms））。指定できる範囲は 0 ～ 999 です。 • *day*：モニターリング間隔（日数）。指定できる範囲は 0 ～ 20 です。
ステップ 4	**diagnostic monitor syslog** 例： `Device(config)# diagnostic monitor syslog`	（任意）ヘルスモニター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 threshold switch** *number number* **test** {*name* | *test-id* | *test-id-range* | **all**} **failure count** *count* 例： `Device(config)# diagnostic monitor threshold switch 2 test 1 failure count 20`	（任意）ヘルスモニター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。 • *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。

	コマンドまたはアクション	目的
		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 monitor switch***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ー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。
ステップ 8	**show diagnostic** { **content** | **post** | **result** | **schedule** | **status** | **switch** }	（任意）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`	（任意）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ーリングおよびメンテナンス

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 *23 :* 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 content switch [*number* | **all**]	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。 **switch** [*number* | **all**] パラメータは、スタック構成スイッチだけでサポートされます。
show diagnostic status	現在実行中の診断テストを表示します。.
show diagnostic result switch [*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。 **switch** [*number* | **all**] パラメータは、スタック構成スイッチだけでサポートされます。
show diagnostic switch [*number* | **all**] [**detail**]	オンライン診断テストの結果を表示します。 **switch** [*number* | **all**] パラメータは、スタック構成スイッチだけでサポートされます。
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。 **switch** [*number* | **all**] パラメータは、スタック構成スイッチだけでサポートされます。
show diagnostic post	POST 結果を表示します（出力は **show post** コマンドの出力と同じ）。
show diagnostic events {**event-type** | **module**}	テスト結果に基づいて、エラー、情報、警告などの診断イベントを表示します。
show diagnostic description module [**number**] **test** { **name** | **test-id** | **all** }	個々のテストまたはすべてのテストの結果について簡単な説明を表示します。

オンライン診断のコンフィギュレーション例

次のセクションでは、オンライン診断の設定例を示します。

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 DiagPOETest
```

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：ヘルスモニターリングテストの設定

次に、ヘルス モニターリング 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#
```

オンライン診断に関する追加情報

関連資料

関連項目	マニュアル タイトル
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 9300* シリーズ スイッチ）

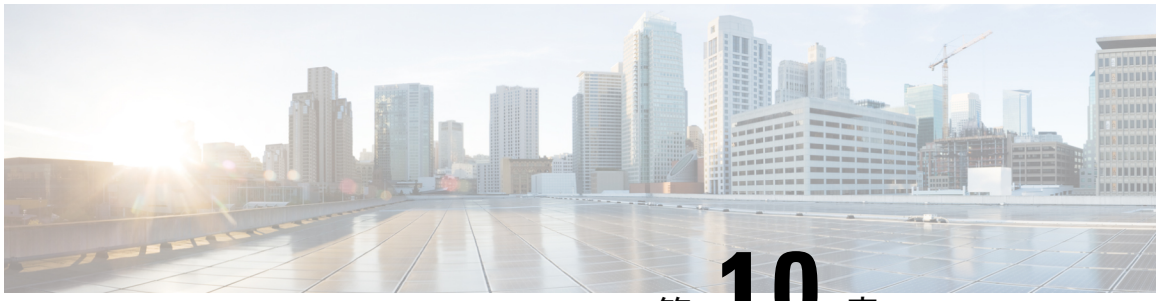
オンライン診断設定の機能情報

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

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

リリース	機能	機能情報
Cisco IOS XE Everest 16.5.1a	オンライン診断	オンライン診断機能を使用すると、デバイスをアクティブ ネットワークに接続したまま、デバイスのハードウェア機能をテストして確認できます。

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



第 10 章

整合性チェッカー

整合性チェッカーの制限事項

整合性チェッカーには次の制限事項があります。

- 整合性チェッカーは CPU 集約型です。短すぎる間隔でチェッカーを実行することは推奨されません。
- レガシー整合性チェッカーはスナップショットをサポートしていません。したがって、以前の実行は表示できません。
- すでに実行中の整合性チェッカーを停止/中止するコマンドはありません。
- 転送エンジンのハードウェアエントリの検証は部分的に実装されます。プログラミングの失敗のみを検出して報告できます。
- レイヤ 2 MAC 整合性チェッカーは、ソフトウェアコピーを使用してハードウェアの MAC アドレスを検証できます。
- 整合性チェッカーは、すべてのケースで誤検出を減らすように設計されています。ただし、次のシナリオではまれに誤検出が報告されることがあります。
  - 大規模なテーブル状態の変更（クリア、再学習など）。
  - 整合性チェッカーの実行中に、他の機能が原因で CPU 使用率が非常に高くなった場合。整合性チェッカーが、CPU使用率が高いプロセスの不整合を報告する場合があります。

整合性チェッカーに関する情報

整合性チェッカーの概要

整合性チェッカーは、ソフトウェアおよびハードウェア内のさまざまなテーブルの状態に関する情報を収集します。ソフトウェアの状態とハードウェアの状態を比較します。不整合がある場合は、ただちに問題にフラグが付けられます。これにより、後のトラブルシューティングの時間を短縮できます。整合性チェッカーは、基本的なトラブルシューティングを補足するもので、ソフトウェアテーブルとハードウェアテーブル間の不整合な状態がネットワークの問題を引き起こしているシナリオを特定するのに役立ちます。これにより、問題を解決するための平均時間が短縮されます。

実装できる整合性チェッカーには、次の 2 つのタイプがあります。

- レガシー整合性チェッカー：コントロールプレーンから転送エンジン（またはハードウェアコピー）へのエントリの検証をサポートします。
- エンドツーエンドの整合性チェッカー：コントロールプレーンから、エントリの配布と処理に関係するすべてのプロセス、および転送エンジンのハードウェアコピーまでのソフトウェアエントリの検証をサポートします。

エンドツーエンドの整合性チェッカー

エンドツーエンド（E2E）の整合性チェッカーは、フルスキャンと単一エントリをサポートしており、手動で開始するか、GOLD診断で実行する必要があります。整合性チェッカーは、転送プロセスのエントリに整合性がないという問題を特定し、デバッグを高速化するためのコマンドを使用して、単一エントリに対して開始できます。

整合性チェッカーが開始されるたびに、runIDが提供されます。runIDを使用して、そのステータス、概要、詳細を表示できます。以前の実行結果を確認するため、直近の 5 つのスナップショットをいつでも入手できます。

E2E 整合性チェッカーは、次の機能を実行します。

- すべてのモジュールのソフトウェアテーブル/プロセス（転送マネージャ RP、転送マネージャ FP、および FED）への IOS エントリを検証します。
- さまざまな不整合（エントリの不整合、エントリの欠落、古いエントリ）を報告し、syslogを送信して管理者に警告します。
- 迅速な障害の特定に役立ちます。
- 矛盾するエントリと関連データを記録します。
- 整合性チェッカーは、実際のエントリとともに依存オブジェクトを検証できる再帰単一エントリチェックをサポートしています（つまり、N個の発信インターフェイスを持つレイヤ 3 マルチキャストを、OIF プログラミング、OIF の隣接関係検証などとともに、マルチキャストエントリについて検証できます）。

• テーブルの合計エントリ数に関係なく、メモリ使用量は一定です。

（注） 整合性チェッカーは CPU 使用率にバインドされているため、プロセス全体でテーブルを検証している間に設定された値を超えることはありません。

整合性チェッカーでサポートされる機能

整合性チェッカーでは次の機能がサポートされています。

- レガシー整合性チェッカー
  - **レイヤ 2 MAC 整合性チェッカー**：この整合性チェッカーは、IOS エントリから FED ソフトウェアエントリを検証します。また、ハードウェアテーブルの MAC アドレスを検証します。
  - **レイヤ 3 FMANFP エントリ整合性チェッカー**：この整合性チェッカーは、転送マネージャ FP プロセスのレイヤ 2、レイヤ 3、およびマルチキャストオブジェクトのステータスを検証します。これには、古いオブジェクトと長期間保留中のオブジェクトが含まれます。
- E2E 整合性チェッカー
  - **レイヤ 2 マルチキャスト整合性チェッカー**：この整合性チェッカーは、IOS レイヤ 2 マルチキャスト IGMP/MLD VLAN、転送マネージャ FP ソフトウェアエントリへのグループエントリ、FED ソフトウェアエントリ、および FED ハードウェア プログラミング エラーを検証します。

整合性チェッカーの実行

次の表は、さまざまな整合性チェッカーを実行するコマンドを示します。

コマンド	目的
show consistency-checker l2	レイヤ 2 転送テーブルで consistency-checker を実行します。
show consistency-checker l3	レイヤ 3 転送テーブルで consistency-checker を実行します。
show consistency-checker mcast l2m	レイヤ 2 マルチキャスト転送テーブルで consistency-checker を実行します。
show consistency-checker objects	オブジェクトでエンドツーエンドの consistency-checker を実行します。

コマンド	目的
show consistency-checker run-id *run-id*	実行 ID ごとにエンドツーエンドの consistency-checker を実行します。
show consistency-checker switch	指定したスイッチで consistency-checker を実行します。

整合性チェッカーの出力例

次に、整合性チェッカーがフルスキャンを実行する **show consistency-checker mcast l2m** コマンドの出力例を示します。

```
Device# show consistency-checker mcast l2m start all
L2 multicast Full scan started. Run_id: 2
Use 'show consistency-checker run-id 2 status' for completion status.

Device#
*Feb 17 06:19:14.889: %FED_CCK_ERRMSG-4-INCONSISTENCY_FOUND: F0/0: fed: Consistency
Checker(CCK) detected inconsistency for l2m_vlan. Check 'show consistency run-id 2
detail'.
*Feb 17 06:19:14.890: %FED_CCK_ERRMSG-4-INCONSISTENCY_FOUND: F0/0: fed: Consistency
Checker(CCK) detected inconsistency for l2m_group. Check 'show consistency run-id 2
detail'.
Device#
*Feb 17 06:19:19.432: %IOSXE_FMANRP_CCK-6-FMANRP_COMPLETED: Consistency Check for Run-Id
 2 is completed. Check 'show consistency-checker run-id 2'.
Device#
Device# show consistency-checker run-id 2 status
Process: IOSD
  Object-Type      Status           Time(sec)     Exceptions
  l2m_vlan         Completed        13            No
  l2m_group        Completed        13            No

Process: FMAN-FP
  Object-Type      Status           Time(sec)     State
  l2m_vlan         Completed        9             Consistent
  l2m_group        Completed        9             Consistent

Process: FED
  Object-Type      Status           Time(sec)     State
  l2m_vlan         Completed        9             Inconsistent
  l2m_group        Completed        9             Inconsistent

Device#
Device# show consistency-checker run-id 2
Process: IOSD
  Object-Type    Start-time              Entries       Exceptions
  l2m_vlan       2021/02/17 06:19:05          22                0
  l2m_group      2021/02/17 06:19:05          24                0

Process: FMAN-FP
  *Statistics(A/I/M/S/Oth): Actual/Inherited/Missing/Stale/Others

  Object-Type    Start-time              State             A/  I/  M/  S/Oth
  l2m_vlan       2021/02/17 06:19:05     Consistent        0/  0/  0/  0/  0
  l2m_group      2021/02/17 06:19:05     Consistent        0/  0/  0/  0/  0
```

```
Process: FED
  *Statistics(A/I/M/S/HW/Oth): Actual/Inherited/Missing/Stale/Hardware/Others

  Object-Type    Start-time              State              A/  I/  M/  S/ HW/Oth
  l2m_vlan       2021/02/17 06:19:05     Inconsistent       1/  0/  0/168/  0/  0
  l2m_group      2021/02/17 06:19:05     Inconsistent       4/  0/  2/  0/  0/  0


Device#
Device# show consistency-checker run-id 2 detail
Process: IOSD

Process: FMAN-FP

Process: FED
  Object-Type:l2m_vlan   Start-time:2021/02/17 06:19:05
    Status:Completed   State:Inconsistent
    Key/data                                Reason
    (Ipv4, vlan: 768)                       Stale
      snoop:off stp_tcn:off flood:off pimsn:off
    (Ipv4, vlan: 769)                       Stale
      snoop:off stp_tcn:off flood:off pimsn:off
    (Ipv6, vlan: 900)                       Inconsistent
      snoop:on stp_tcn:on flood:on pimsn:off
    (Ipv6, vlan: 767)                       Stale
      snoop:off stp_tcn:off flood:off pimsn:off

  Object-Type:l2m_group   Start-time:2021/02/17 06:19:05
    Status:Completed   State:Inconsistent
    Key/data                                Reason
    (Ipv4, vlan:100 (*,227.0.0.0))          Inconsistent
      Group ports: total entries: 0
    (Ipv4, vlan:100 (*,227.1.0.0))          Missing

Device#
```

次に、整合性チェッカーが再帰的な単一エントリスキャンを実行する **show consistency-checker mcast l2m** コマンドの出力例を示します。

```
Device# show consistency-checker mcast l2m start vlan 900 229.1.1.1 recursive
Single entry scan started with Run_id: 2

*Feb 17 06:54:09.880: %IOSXE_FMANRP_CCK-6-FMANRP_COMPLETED: Consistency Check for Run-Id
 2 is completed.
Check 'show consistency-checker run-id 2'.
Device#
Device# show consistency-checker run-id 2
Process: IOSD
  Object-Type    Start-time              Entries       Exceptions
  l2m_vlan       2021/02/17 06:54:01           1                0
  l2m_group      2021/02/17 06:54:01           1                0

Process: FMAN-FP
  *Statistics(A/I/M/S/O): Actual/Inherited/Missing/Stale/Others

  Object-Type    Start-time              State            A / I / M / S / O
  l2m_vlan       1970/01/01 00:10:03     Consistent        0/  0/  0/  0/  0
  l2m_group      1970/01/01 00:10:03     Consistent        0/  0/  0/  0/  0


Process: FED
  *Statistics(A/I/M/S/HW/O): Actual/Inherited/Missing/Stale/Hardware/Others

  Object-Type    Start-time              State             A / I / M / S / HW/ O
```

```
  l2m_vlan       2021/02/17 06:54:01     Inconsistent       1/  0/  0/  0/  0/  0
  l2m_group      2021/02/17 06:54:01     Inconsistent       0/  1/  0/  0/  0/  0

Device#
Device# show consistency-checker run-id 2 detail
Process: IOSD
  Object-Type:l2m_vlan   Start-time:2021/02/17 06:54:01
    Key/data                                Reason
    (Ipv4, vlan:900)                        Success
      snoop:on stp_tcn:off flood:off pimsn:off

  Object-Type:l2m_group   Start-time:2021/02/17 06:54:01
    Key/data                                Reason
    (Ipv4, vlan:900, (*,229.1.1.1))         Success
      Twe1/0/5


Process: FMAN-FP

Process: FED
  Object-Type:l2m_group   Start-time:2021/02/17 06:54:01
    Status:Completed   State:Inconsistent
    Key/data                                Reason
    (Ipv4, vlan:900 (*,229.1.1.1))          Inherited
      Group ports: total entries: 1
        TwentyFiveGigE1/0/5

      ---------------Recursion-level-1-----------------
      Object-Type:l2m_vlan   Start-time:2021/02/17 06:54:01
        Status:Completed   State:Inconsistent
        Key/data                                Reason
        (Ipv4, vlan: 900)                       Inconsistent
          snoop:on stp_tcn:off flood:on pimsn:off

Device#
```

次に、整合性チェッカーがオブジェクトのスキャンを実行する **show consistency-checker objects** コマンドの出力例を示します。

```
Device# show consistency-checker objects l2m_group
Process: IOSD
  Run-id    Start-time              Exception
  1         2021/02/17 05:20:42     0
  2         2021/02/17 06:19:05     0

Process: FMAN-FP
  *Statistics(A/I/M/S/Oth): Actual/Inherited/Missing/Stale/Others

  Run-id    Start-time              State           A/  I/  M/  S/Oth
  1         2021/02/17 05:20:42     Consistent      0/  0/  0/  0/  0
  2         2021/02/17 06:19:05     Consistent      0/  0/  0/  0/  0

Process: FED
  *Statistics(A/I/M/S/HW/Oth): Actual/Inherited/Missing/Stale/Hardware/Others

  Run-id    Start-time              State           A/  I/  M/  S/ HW/Oth
  1         2021/02/17 05:20:42     Consistent      0/  0/  0/  0/  0/  0
  2         2021/02/17 06:19:05     Inconsistent    4/  0/  2/  0/  0/  0

Device#
Stark#sh consistency-checker run 2 detail
Process: IOSD
  Object-Type:l2m_vlan   Start-time:2021/02/17 06:54:01
    Key/data                                Reason
```

```
    (Ipv4, vlan:900)                          Success
      snoop:on stp_tcn:off flood:off pimsn:off

  Object-Type:l2m_group   Start-time:2021/02/17 06:54:01
    Key/data                                  Reason
    (Ipv4, vlan:900, (*,229.1.1.1))           Success
      Twe1/0/5


Process: FMAN-FP

Process: FED
  Object-Type:l2m_group   Start-time:2021/02/17 06:54:01
    Status:Completed   State:Inconsistent
    Key/data                                  Reason
    (Ipv4, vlan:900 (*,229.1.1.1))            Inherited
      Group ports: total entries: 1
        TwentyFiveGigE1/0/5

      ---------------Recursion-level-1-----------------
      Object-Type:l2m_vlan   Start-time:2021/02/17 06:54:01
        Status:Completed   State:Inconsistent
        Key/data                                    Reason
        (Ipv4, vlan: 900)                           Inconsistent
          snoop:on stp_tcn:off flood:on pimsn:off

Device# show consistency-checker objects l2m_group 2 detail
Process: IOSD

Process: FMAN-FP

Process: FED
  Object-Type:l2m_group   Start-time:2021/02/17 06:19:05
    Status:Completed   State:Inconsistent
    Key/data                                  Reason
    (Ipv4, vlan:100 (*,227.0.0.0))            Inconsistent
      Group ports: total entries: 0
    (Ipv4, vlan:100 (*,227.1.0.0))            Missing
    (Ipv4, vlan:100 (*,227.0.0.1))            Inconsistent
      Group ports: total entries: 0
    (Ipv4, vlan:100 (*,227.1.0.1))            Missing
    (Ipv4, vlan:100 (*,227.0.0.2))            Inconsistent
      Group ports: total entries: 0
    (Ipv4, vlan:100 (*,227.0.0.3))            Inconsistent
      Group ports: total entries: 0


Device#
```

整合性チェッカーの機能履歴

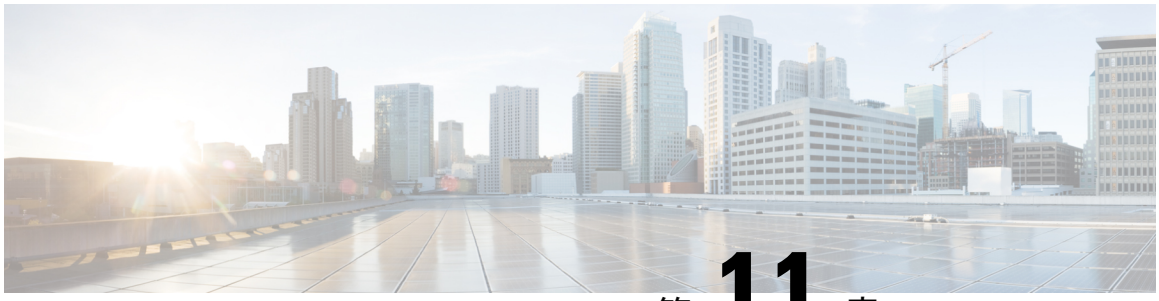
次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

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

リリース	機能	機能情報
Cisco IOS XE Amsterdam 17.3.1	整合性チェッカー	整合性チェッカーは、ソフトウェアおよびハードウェア内のさまざまなテーブルの状態に関する情報を収集し、不整合が検出されるとすぐにフラグを付けます。これは、基本的なトラブルシューティングを補足するもので、ソフトウェアテーブルとハードウェアテーブル間の不整合な状態がネットワークの問題を引き起こしているシナリオを特定するのに役立ちます。これにより、問題を解決するための平均時間が短縮されます。
Cisco IOS XE Bengaluru 17.6.1	整合性チェッカー	この機能が拡張され、マルチキャスト整合性チェッカーが導入されました。**mcast**、**objects**、**run-id** のキーワードが **show consistency-checker** コマンドに追加されました。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator にアクセスするには、https://cfnng.cisco.com/にアクセスします。

http://www.cisco.com/go/cfn。



第 11 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

- ユーザーには、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 モードでのみ使用可能であり機能します。
- 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、シスコ製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。その場合は、**configure terminal** 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（395 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、ファイルサーバー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 サーバー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 サーバー（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 サーバー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）サーバー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、ユーザー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 （419 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）
  - **usbflash1:**（外部 usbflash ファイル システム）

ネットワークサーバー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サーバー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。
- 別のデバイスの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、そのデバイス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。ファイルを新しいデバイスにコピーすると、ファイル全体を再作成しないで、関連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まで、既存の実行コンフィギュレーションは消去されません。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ー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。

サーバー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、次のセクションで説明するタスクを実行します。

使用するプロトコルは、使用中のサーバー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。

デバイスから TFTP サーバー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サーバー上にダミー 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 サーバー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 サーバー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、ユーザー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ーにあるファイル 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 サーバー（またはデーモン）を利用します。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ー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ーへのアクセスだけです（ほとんどの UNIX システムが rsh をサポートしています）。ある場所から別の場所にファイルをコピーするため、コピー元のファイルに対する読み取り権限とコピー先の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 サーバー（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 システムのユーザーがデバイスからまたはデバイス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 ユーザーがデバイスとの間でファイルをコピーできるように Cisco IOS ソフトウェアを設定するには、**ip rcmd rcp-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。

機能制限

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 ユーザー名をサーバー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ー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ー名を送信します。

1. **copy** EXEC コマンドで指定されたユーザー名（ユーザー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ー名（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 ユーザー名。たとえば、ユーザーが 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、リモートユーザー名として Telnet ユーザー名が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 サーバー上にリモート ユーザー名のアカウントを定義する必要があります。このサーバー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、サーバー上のリモート ユーザー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 イメージがサーバー上のユーザーのホーム ディレクトリにある場合は、そのユーザーの名前をリモート ユーザー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ー名を指定します。（rcmd は、スーパーユーザー 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ー名を指定する場合は、**copy** コマンド内でユーザー名を指定します。

サーバーに書き込む場合、デバイス上のユーザーからの RCP 書き込み要求を受け入れるように、RCP サーバー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 サーバー上のリモートユーザー用の .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 device1.example.com に変換される場合、RCP サーバー上の 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 ユーザー名に関する要件

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 ユーザー名をサーバー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ー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ー名を送信します。

1. **copy** EXEC コマンドで指定されたユーザー名（ユーザー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ー名（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ー名。たとえば、ユーザーが 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、リモートユーザー名として Telnet ユーザー名が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 サーバー上にリモート ユーザー名のアカウントを定義する必要があります。このサーバー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、サーバー上のリモート ユーザー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ー上のユーザーのホームディレクトリにある場合は、そのユーザーの名前をリモート ユーザー名として指定します。

詳細については、ご使用の RCP サーバーのマニュアルを参照してください。

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

FTP ユーザー名およびパスワードの概要

（注） パスワードに特殊文字「@」を含めることはできません。文字「@」を使用すると、コピーでサーバーの IP アドレスを解析できません。

FTPプロトコルでは、FTP要求ごとにリモートユーザー名およびパスワードを、クライアントがサーバー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ー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ー名を送信します。

1. **copy** EXEC コマンドで指定されたユーザー名（ユーザー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ー名（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*@*devicename*.*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に関連付けられたユーザー名、*devicename* は設定済みのホスト名、*domain* はデバイスのドメインです。

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

このサーバー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、サーバー上のユーザー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 イメージがサーバー上のユーザーの

ホーム ディレクトリにある場合は、そのユーザーの名前をリモート ユーザー名として指定します。

詳細については、ご使用の FTP サーバーのマニュアルを参照してください。

すべてのコピー操作に使用するユーザー名およびパスワードを指定するには、**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 コマンドを使用します。特定のコピー操作にのみ使用するユーザー名を指定する場合は、**copy** EXEC コマンド内でユーザー名を指定します。

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 サーバーに設定をコピーし、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（419 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ーに格納しておき、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 サーバーを使用して大規模な設定を格納するには、デバイスから TFTP サーバーへのコンフィギュレーション ファイルのコピー （397 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 （394 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 サーバー上にあり、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 モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**show boot** 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。

	コマンドまたはアクション	目的
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	**show running-config** 例： `Device# show running-config`	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 system:running-config** コマンドのコマンドエイリアスです）。
ステップ 5	**show startup-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です）。 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、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 **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）サーバー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、ユーザー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 モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル設定モードを開始します。
ステップ 3	**configuration command** 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	**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 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、

	コマンドまたはアクション	目的
		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を設定しています。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** コマンドを使用して、デバイス名を device から new_name に変更しています。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。

デバイスから TFTP サーバー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 サーバー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 モードを有効にします。 • パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
ステップ 2	**copy system:running-config tftp:** [[[*//location*]*/directory*]*/filename*] 例： `Device# copy system:running-config tftp: //server1/topdir/file10`	TFTP サーバー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。
ステップ 3	**copy nvram:startup-config tftp:** [[[*//location*]*/directory*]*/filename*] 例： `Device# copy nvram:startup-config tftp: //server1/1stdir/file10`	TFTP サーバーへスタートアップ 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、デバイスからから TFTP サーバー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。

デバイスから RCP サーバー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 サーバーへスタートアップ 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 モードを有効にします。 • パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル設定モードを開始します。
ステップ 3	**ip rcmd remote-username** *username* 例： `Device(config)# ip rcmd remote-username NetAdmin1`	（任意）デフォルトのリモートユーザー名を変更します。
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 サーバー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 サーバー上に格納されるように指定します。

例

RCP サーバー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 サーバー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、サーバー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 サーバー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 サーバーへスタートアップ 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 モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	デバイスでグローバル 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	**ip ftp username** *username* 例： `Device(config)# ip ftp username NetAdmin1`	（任意）デフォルトのリモート ユーザー名を指定します。
ステップ 4	**ip ftp password** *password* 例：	（任意）デフォルトのパスワードを指定します。

	コマンドまたはアクション	目的
	`Device(config)# ip ftp password adminpassword`	
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 ユーザー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 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。 • **copy system:running-config ftp:** [[[//[*username* [**:***password*]**@**]*location*]/*directory*]/*filename*] または • **copy nvram:startup-config ftp:** [[[//[*username* [**:***password*]**@**]*location*]/*directory*]/*filename*] 例： `Device# copy system:running-config ftp:`	FTP サーバー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 サーバー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 サーバー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、サーバー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 サーバー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 サーバー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、以下のタスクを実行します。

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 モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**copy tftp:** [[[*//location*]*/directory*]*/filename*] **system:running-config** 例： `Device# copy tftp://server1/dir10/datasource system:running-config`	TFTP サーバー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	**copy tftp:** [[[*//location*]*/directory*]*/filename*] **nvram:startup-config** 例： `Device# copy tftp://server1/dir10/datasource nvram:startup-config`	TFTP サーバー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	**copy tftp:** [[[*//location*]*/directory*]*/filename*] **flash-[n]:/directory/startup-config** 例： `Device# copy`	TFTP サーバー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。

	コマンドまたはアクション	目的
	`tftp://server1/dir10/datasource flash:startup-config`	

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 サーバー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 サーバー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 モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 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 ユーザー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	**ip rcmd remote-username** *username* 例：	（任意）リモート ユーザー名を指定します。

	コマンドまたはアクション	目的
	`Device(config)# ip rcmd remote-username NetAdmin1`	
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。この手順は、デフォルトのリモート ユーザー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 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 サーバー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 のリモートサーバー上の 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て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 ユーザー名 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 サーバー上の 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 サーバー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 サーバー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 モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 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 ユーザー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	**ip ftp username** *username* 例： `Device(config)# ip ftp username NetAdmin1`	（任意）デフォルトのリモートユーザー名を指定します。
ステップ 4	**ip ftp password** *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 ユーザー名

	コマンドまたはアクション	目的
	`Device(config)# end`	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 6	次のいずれかを実行します。 • **copy ftp:** [[[//[*username*[**:***password*]**@**]*location*] **/***directory*]**/***filename*]**system:running-config** • **copy ftp:** [[[*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 サーバー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 という名前のホスト 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 のリモートサーバー上の 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て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 ユーザー名 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 サーバー上の 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 モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル設定モードを開始します。
ステップ 3	**service compress-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。 • **configure terminal** 例：	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とすると、次のエラー メッセージが表示されます。

	コマンドまたはアクション	目的
	`Device# configure terminal`	「[buffer overflow - *file-size* /*buffer-size* bytes]。」
ステップ 6	**copy system:running-config nvram: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 モードを有効にします。 • パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
ステップ 2	**copy nvram: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	**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 4	**boot config flash-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] 」 • **configure terminal** 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	**copy system:running-config nvram: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 サーバーを使用して、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 モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**copy system: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ーに保存します。
ステップ 3	**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 4	**boot network** {**ftp:**[[[//[*username* [**:***password*]**@**]*location*]/*directory*]/*filename*] | **rcp:**[[[//[*username***@**]*location*]/*directory*]/*filename*] | **tftp:**[[[//*location*]/*directory*]/*filename*]} 例： `Device(config)# boot network ftp://user1:guessme@example.com/dir10/file1`	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 サーバー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
ステップ 5	**service config** 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	**copy system:running-config nvram: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 モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	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： `Device# copy usbflash0:4:ios-upgrade-1 nvram:startup-config`	• 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。

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ことで、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 モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**show** *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*] 例： `Device# copy flash: usbflash0:`	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。たとえば、**copy usbflash0: usbflash0:** コマンドが無効です。

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

FTP サーバーからフラッシュ 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 サーバーからフラッシュ 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：	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
	`Device> enable`	
ステップ 2	**configure terminal** 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 ユーザー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	**ip ftp username** *username* 例： `Device(config)# ip ftp username Admin01`	（任意）リモート ユーザー名を指定します。
ステップ 4	**ip ftp password** *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 ユーザー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	**copy ftp:** [[*//location*]*/directory*]*/bundle_name* **flash:** 例： `Device>copy ftp:/cat9k_iosxe.16.11.01.SPA.bin flash:`	FTP を使用してネットワーク サーバーからフラッシュ 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 サーバーからフラッシュ 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 サーバーからフラッシュ 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 モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 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 ユーザー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	**ip rcmd remote-username** *username* 例： `Device(config)# ip rcmd remote-username Admin01`	（任意）リモート ユーザー名を指定します。
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ー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	**copy rcp:** [[[//[*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 サーバーからフラッシュ 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 **file prompt** コマンドの現在の設定によって異なります。

TFTP サーバーからフラッシュ 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 サーバーからフラッシュ 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 モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**copy tftp:** [[[*//location*]*/directory*]/bundle_name **flash:** 例： `Device#` `copy` `tftp://a3k.ca-universal-9.SPA.03.12.02.EZP.150-12.02.EZP.150-12.02.EZP.bin` ` flash:`	TFTP サーバーからフラッシュ 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 サーバーから 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：	特権 EXEC モードを有効にします。

	コマンドまたはアクション	目的
	`Device> enable`	• パスワードを入力します（要求された場合）。
ステップ 2	**configure memory** 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 モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**erase nvram** 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、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が、デバイスにより削除されます。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

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 モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**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 モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**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	**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 4	**boot config** *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	**copy system:running-config nvram: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	**show boot** 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の場所に関係なく、スタートアップ コンフィギュレーションが表示されます。**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コンフィギュレーションを保存した場合、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 環境変数により指定された場所に保存され、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は、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は、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進めます。

（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「削除済み」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、領域を解放してください。

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、次のセクションで説明するタスクを少なくとも 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

起動時にサーバーからネットワーク 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 モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル設定モードを開始します。
ステップ 3	**boot network** {**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。 • 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、

	コマンドまたはアクション	目的
	`Device(config)# boot network tftp:hostfile1`	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 サーバー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	**service config** 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

起動時にサーバーからホスト 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：	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
	`Device> enable`	
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル設定モードを開始します。
ステップ 3	**boot host** {**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 device-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 サーバー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	**service config** 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。

	コマンドまたはアクション	目的
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。

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
```

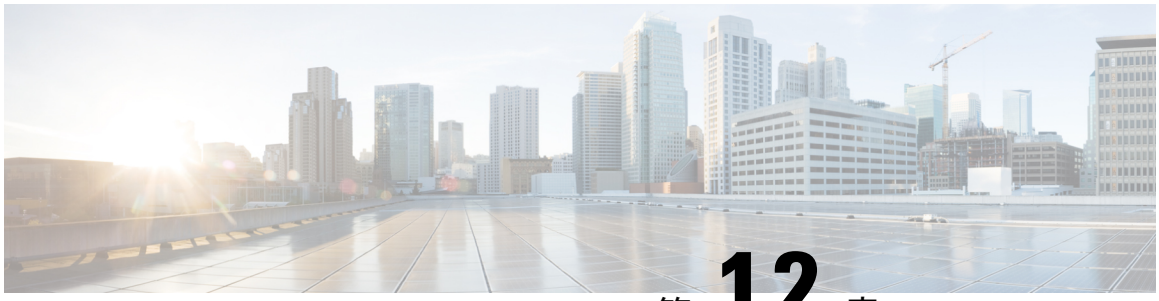
コンフィギュレーション ファイルの管理の機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

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

リリース	機能	機能情報
Cisco IOS XE Everest 16.5.1a	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 ファイルには、シスコ製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。

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



第 12 章

セキュア コピー

このドキュメントでは、セキュアコピー（SCP）サーバー側機能用にシスコデバイスを設定する手順について説明します。

セキュア コピーの前提条件

- デバイス上でセキュアシェル（SSH）、認証、および許可を設定します。
- Secure Copy Protocol（SCP）は SSH を使用してセキュアな転送を実行するため、デバイスには Rivest、Shamir、Adelman（RSA）キーのペアが必要です。

Secure Copy に関する情報

Secure Copy 機能は、スイッチの設定やイメージファイルのコピーにセキュアな認証方式を提供します。Secure Copy Protocol（SCP）は、セキュアシェル（SSH）、アプリケーション、および Berkeley r ツールのセキュアな代替手段を提供するプロトコルに依存します。

SCP は一連の Berkeley の r ツール（Berkeley 大学独自のネットワーキング アプリケーション セット）に基づいて設計されているため、その動作内容は Remote Copy Protocol（RCP）と類似しています。ただし、SCP は SSH のセキュリティに対応している点は除きます。加えて、SCP では、ユーザーが正しい権限レベルを持っていることをデバイス上で判断できるように、認証、許可、およびアカウンティング（AAA）を設定する必要があります。

SCP を使用すると、**copy** コマンドを使用して Cisco IOS ファイルシステム（Cisco IFS）内の任意のファイルのコピーをデバイスとの間で実行できるのは、特権レベルが 15 のユーザーのみ

になります。許可された管理者はワークステーションからこの操作を実行することもできます。

（注）
- pscp.exe ファイルを使用している場合は、SCP オプションを有効にします。
- SSH を機能させるには、RSA 公開キーと秘密キーのペアをデバイスで設定する必要があります。

セキュアコピーのパフォーマンス向上

SSH一括データ転送モードを使用すると、クライアントまたはサーバーの容量で動作するSCPのスループットパフォーマンスを向上させることができます。このモードはデフォルトでは無効になっていますが、**ip ssh bulk-mode** 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有効にすることができます。一括モードウィンドウサイズが設定されている場合、TCP選択的確認応答（SACK）はデフォルトでイネーブルになります。

（注）このコマンドは、大きなファイルを転送する場合にのみ有効にし、ファイル転送の完了後に無効にすることをお勧めします。

デフォルトの一括モードウィンドウサイズである 128 KB は、ほとんどのネットワーク設定で大きなファイルをコピーするのに最適ですが、ラウンドトリップ時間（RTT）が広帯域高遅延ネットワークでは、128 KB では不十分です。**ip ssh bulk-mode** *window-size* コマンドを使用して一括モードウィンドウサイズを設定することで、最適な SCP スループットパフォーマンスをイネーブルにできます。たとえば、理想的なラボテスト環境では、200 ミリ秒のラウンドトリップ時間設定で 2 MB のウィンドウサイズを設定すると、デフォルトの 128 KB のウィンドウサイズと比較して、スループットパフォーマンスが約 500% 向上します。

一括モードウィンドウサイズは、ネットワーク帯域幅遅延積（つまり、使用可能な合計帯域幅（bps）およびラウンドトリップ時間（秒）の乗数）に従って設定する必要があります。ウィンドウサイズが大きくなると CPU 使用率が増加する可能性があるため、適切なウィンドウサイズを選択してバランスを取ります。

セキュア コピーの設定方法

ここでは、セキュアコピーの設定作業について説明します。

セキュアコピーの設定

シスコデバイスに SCP サーバー側機能の設定をするには、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 モードを有効にします。 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	**aaa new-model** 例： `Device(config)# aaa new-model`	ログイン時の AAA 認証を設定します。
ステップ 4	**aaa authentication login** {**default** | *list-name*} *method1* [*method2...*] 例： `Device(config)# aaa authentication login default group tacacs+`	AAA アクセス コントロール システムをイネーブルにします。
ステップ 5	**username** *name* [**privilege** *level*] **password** *encryption-type encrypted-password* 例： `Device(config)# username superuser privilege 2 password 0 superpassword`	ユーザ名をベースとした認証システムを構築します。 （注） TACACS+ や RADIUS などのネットワークベースの認証メカニズムが設定されている場合は、この手順を省略できます。
ステップ 6	**ip scp server enable** 例： `Device(config)# ip scp server enable`	SCP サーバ側機能を有効にします。
ステップ 7	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 8	**debug ip scp** 例： `Device# debug ip scp`	（任意）SCP 認証問題を解決します。

SSH サーバーでのセキュアコピーのイネーブル化

次のタスクでは、SCPのサーバー側機能の設定方法を示します。このタスクは、デバイスでリモートのワークステーションからファイルを安全にコピーできる一般的な設定を示しています。

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 モードを有効にします。 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	**aaa new-model** 例： `Device(config)# aaa new-model`	認証、許可、アカウンティング（AAA）アクセス コントロール モデルをイネーブルにします。
ステップ 4	**aaa authentication login default local** 例： `Device(config)# aaa authentication login default local`	ログイン時の認証にローカルのユーザー名データベースを使用するように AAA 認証を設定します。
ステップ 5	**aaa authorization exec default local** 例： `Device(config)# aaa authorization exec default local`	ユーザーアクセスを制限するパラメータをネットワークに設定します。許可を実行し、ユーザー ID で特権 EXEC シェルの実行を許可するかどうかを定義します。その後、システムで許可にローカルデータベースを使用する必要があることを指定します。
ステップ 6	**username** *name* **privilege** *privilege-level* **password** *password* 例： `Device(config)# username samplename privilege 15 password password1`	ユーザー名ベースの認証システムを確立し、ユーザー名、権限レベル、および非暗号化パスワードを指定します。 （注） *privilege-level* 引数に必要な最小値は 15 です。権限レベルが 15 未満の場合、接続が切断されます。

	コマンドまたはアクション	目的
ステップ 7	**ip ssh time-out** *seconds* 例： `Device(config)# ip ssh time-out 120`	デバイスが SSH クライアントの応答を待つ時間間隔を、秒単位で設定します。
ステップ 8	**ip ssh authentication-retries** 整数 例： `Device(config)# ip ssh authentication-retries 3`	インターフェイスのリセット後、認証を試行する回数を設定します。
ステップ 9	**ip scp server enable** 例： `Device(config)# ip scp server enable`	デバイスで、リモート ワークステーションから安全にファイルをコピーできるようにします。
ステップ 10	**ip ssh bulk-mode** *window-size* 例： `Device(config)# ip ssh bulk-mode 33107232`	（任意）SSH 一括データ転送モードをイネーブルにして、SCP のスループットパフォーマンスを強化します。
ステップ 11	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 12	**debug ip scp** 例： `Device# debug ip scp`	（任意）SCP 認証の問題に関する診断情報を提供します。

セキュア コピーの設定例

次に、セキュアコピーの設定例を示します。

例：ローカル認証を使用したセキュア コピーの設定

次の例は、セキュアコピーのサーバー側機能の設定方法を示しています。この例では、ローカルに定義されたユーザー名とパスワードを使用します。

```
! AAA authentication and authorization must be configured properly in order for SCP to
work.
Device> enable
```

```
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authentication login default local
Device(config)# aaa authorization exec default local
Device(config)# username user1 privilege 15 password 0 lab
! SSH must be configured and functioning properly.
Device(config)# ip scp server enable
Device(config)# end
```

例：ネットワークベース認証を使用したセキュアコピーのサーバー側の設定

次の例は、ネットワークベースの認証メカニズムを使用したセキュアコピーのサーバー側機能の設定方法を示しています。

```
! AAA authentication and authorization must be configured properly for SCP to work.
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authentication login default group tacacs+
Device(config)# aaa authorization exec default group tacacs+
! SSH must be configured and functioning properly.
Device(config)# ip ssh time-out 120
Device(config)# ip ssh authentication-retries 3
Device(config)# ip scp server enable
Device(config)# end
```

セキュアコピーに関する追加情報

関連資料

関連項目	マニュアル タイトル
セキュア シェル バージョン 1 と 2 のサポート	セキュア シェルの設定

シスコのテクニカル 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

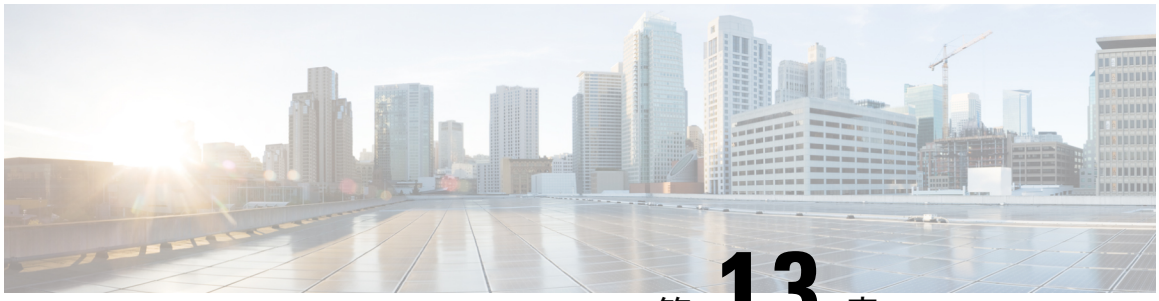
セキュア コピーの機能情報

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

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

リリース	機能	機能情報
Cisco IOS XE Everest 16.5.1a	セキュア コピー	Secure Copy 機能は、デバイス設定またはデバイスイメージファイルをコピーするための安全で認証された方式を提供します。SCP は、SSH、アプリケーション、および Berkeley r ツールのセキュアな代替手段を提供するプロトコルに依存します。 次のコマンドが導入または変更されました。**debug ip scp** および **ip scp server enanle**
Cisco IOS XE Amsterdam 17.2.1	セキュアコピーのパフォーマンス向上	SSH 一括モードを使用すると、特定の最適化により、大量のデータ転送を伴うプロシージャのスループットパフォーマンスを向上できます。このモードは、**ip ssh bulk-mode** 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有効にすることができます。
Cisco IOS XE Bengaluru 17.6.1	大規模な RTT シナリオでのセキュアコピーの改善	大規模な RTT 設定でのセキュアコピーは、**ip ssh bulk-mode** コマンドの *window-size* 変数オプションを使用して設定できます。

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



第 13 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

コンフィギュレーション アーカイブ

Cisco IOS コンフィギュレーション アーカイブは、**configure replace** コマンドにより提供され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、整理、管理を行うことを目的としたメカニズムです。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

configure replace 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。

configure replace 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（**copy running-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。**configure replace** 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、**show archive config differences** 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 **configure replace** *target-url* 特権 EXEC コマンドの代わりに使用する場合、主な相違点として次の点に注意が必要です。

- **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。これに対して、**configure replace** *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。これに対して、**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の動作中、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ー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。

コンフィギュレーション 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。

configure replace 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（**configure replace** *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 デバイスとユーザー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される場合、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シンプルに。
- **configure replace** コマンドを **copy** *source-url* **running-config** コマンドの代用として使用すると、現在の実行コンフィギュレーションにある既存のコマンドが再度適用されないため、効率が向上し、サービス停止のリスクが回避されます。

コンフィギュレーションの置換とロールバックの使用方法

コンフィギュレーション アーカイブの作成

configure replace コマンドを使用するうえで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 ロールバックのシナリオで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 モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル設定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、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** *minutes* 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	**archive config** 例： `Device# archive config`	現在の実行設定ファイルを設定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やロールバック操作の実行

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 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 モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure replace** *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** コマンドを入力しなければならない制限時間（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ー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	**configure revert** { **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	**configure confirm** 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、**configure replace** コマンドの **time** *seconds* キーワードおよび引数が指定されている場合にのみ使用します。
ステップ 5	**exit** 例： `Device# exit`	ユーザー EXEC モードに戻ります。

機能のモニターリングおよびトラブルシューティング

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ー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。パスワードを入力します（要求された場合）。

例：

```
Device> enable
Device#
```

ステップ 2 **show archive**

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 **show archive** 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　debug archive versioning

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ー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　debug archive config timestamp

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、ユーザー 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。**configure replace** 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 ファイルへ復元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 ユーザー 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ともに使用する例を示します。現在の実行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）。

```
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、現在の実行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。

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とロールバックに関するその他の参考資料

関連資料

関連項目	マニュアル タイトル
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 9300* シリーズ スイッチ）

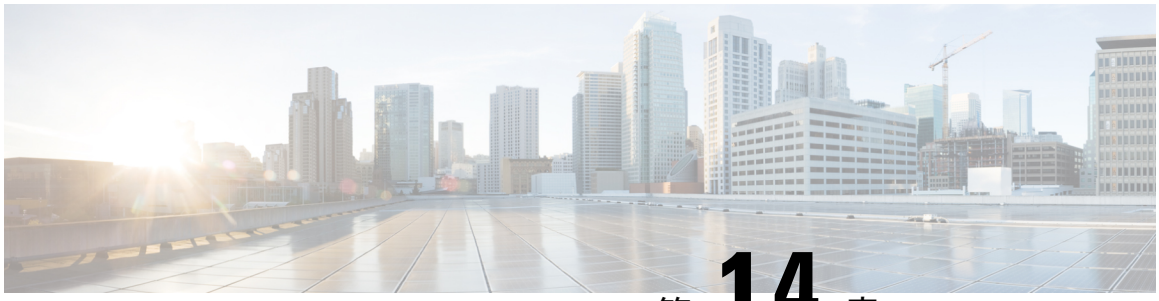
コンフィギュレーションの置換およびコンフィギュレーションのロールバックの機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

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

リリース	機能	機能情報
Cisco IOS XE Everest 16.5.1a	コンフィギュレーションの置換とロールバック	Cisco IOS コンフィギュレーション アーカイブは、**configure replace** コマンドにより提供され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、整理、管理を行うことを目的としたメカニズムです。

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



第 14 章

BIOS 保護

BIOS 保護の概要

BIOS 保護機能により、ゴールデン ROMMON イメージの書き込み保護とセキュアアップグレードが有効になります。ROMMON は、デバイスの電源を投入または再起動したときに、ハードウェアを初期化して Cisco IOS XE ソフトウェアイメージをブートするブートストラッププログラムです。ファームウェア障害を解決するか、新しい機能をサポートするには、ROMMON のアップグレードが必要になることがあります。通常、ROM モニターのアップグレードはまれで、Cisco IOS XE ソフトウェアのアップグレードごとには必要ありません。

BIOS 保護機能がないと、ソフトウェアのアップグレード中に悪意のあるコードによってゴールデン ROMMON が破損する可能性があります。

ROMMON アップグレード

ROMMON イメージは、プライマリ ROMMON およびゴールデン ROMMON として SPI フラッシュデバイスに保存されます。プライマリ ROMMON は、デバイスの電源がオンになるか再起動されるたびに起動します。プライマリ ROMMON が破損した場合、デバイスはゴールデン ROMMON を使用して IOS XE ソフトウェアイメージを起動します。デバイスがプライマリ ROMMON から起動すると、ゴールデン ROMMON はロックされます。BIOS 保護を使用すると、ゴールデン ROMMON は書き込み保護され、フラッシュユーティリティのアップグレードメカニズムを使用してアップグレードすることができません。アクセスポリシーは、FPGA ファームウェアによって管理されます。FPGA は、ゴールデン ROMMON SPI フラッシュデバイスで許可されていない操作（書き込み、消去など）をブロックします。

（注） ゴールデンROMMONアップグレードは、セキュアブートFPGAアップグレードなしでは有効になりません。

プライマリROMMON、プライマリFPGA、およびゴールデンFPGA（セキュアブートFPGA）は、デバイスの起動時に自動的にアップグレードされます。ゴールデンROMMONは、カプセルアップグレードを使用してのみアップグレードできます。

アップグレードプロセスはスタンドアロンシステムと高可用性システムで異なり、以下で説明します。

スタンドアロンシステム

スタンドアロンデバイスでは、デバイスをインストールモードでアップグレードすると、デバイスの起動時にプライマリ ROMMON が自動的にアップグレードされます。ゴールデン ROMMON は、カプセルアップグレードを使用してアップグレードできます。

高可用性および StackWise Virtual システム

高可用性設定のデバイスでは、In-Service Software Upgrade（ISSU）を実行することを推奨します。FPGA のアップグレードは、ISSU の一部として行われます。

リロードを使用してインストールモードでアップグレードを実行する場合は、両方のスーパバイザを同時にリロードしないでください。スタンバイスーパバイザをROMMON状態にして、アクティブスーパバイザを起動します。各スーパバイザでROMMONアップグレードが完了すると、FPGA およびソフトウェアイメージがアップグレードされます。

スタンバイスーパバイザを起動し、スタンバイスーパバイザがアップグレードしてスタンバイホット状態になるようにします。

カプセルアップグレード

カプセルアップグレードでは、ゴールデンROMMONをアップグレードするため、認証後にプライマリ ROMMONによって使用されるセキュアな更新カプセルが作成され、署名されます。セキュアな更新カプセルには、セキュアなフラッシュ証明書が必要です。セキュアなフラッシュ証明書はプロダクトキーを使用して作成され、プライマリROMMONイメージに追加されて更新カプセルの真正性が検証されます。カプセルは、セキュアなフラッシュ証明書とセキュアブート 16 MB フラッシュイメージを使用して作成され、署名されます。

デバイスが起動すると、プライマリ ROMMON がゴールデン ROMMON のカプセルアップグレードをトリガーします。ゴールデンROMMONのカプセルアップグレードを実行するには、特権 EXEC モードで **upgrade rom-monitor capsule golden switch** コマンドを使用します。

カプセルアップグレードでは、次のプロセスが実行されます。

- デバイスは、セキュアブートFPGAアップグレードが有効になっているかどうかを確認します。有効でない場合、プロセスは終了します。

- デバイスは、ブートローダー保護が有効になっているかどうかを確認します。有効でない場合は、プライマリ ROMMON、ゴールデン ROMMON、およびプライマリ FPGA のワンタイムアップグレードが開始されます。
- ブートローダー保護がすでにアクティブになっている場合、IOS はセキュアな更新カプセルをブートフラッシュにコピーし、デバイスを再起動します。
- デバイスが再起動すると、アップグレードを実行するためにセキュアな更新カプセルが選択されます。

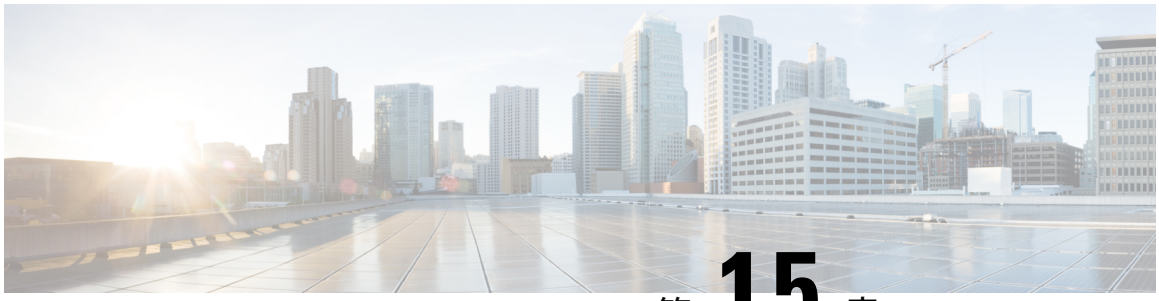
BIOS 保護の機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

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

リリース	機能	機能情報
Cisco IOS XE Gibraltar 16.12.1	BIOS 保護	BIOS 保護機能により、ゴールデン ROMMONイメージの書き込み保護とセキュアアップグレードが有効になります。
Cisco IOS XE Amsterdam 17.1.1	カプセルアップグレード	**upgrade rom-monitor capsule switch active** コマンドを使用したゴールデン ROMMON のカプセルアップグレードのサポートが有効になりました。

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



第 15 章

Extended Fast Software Upgrade の実行

Extended Fast Software Upgrade の前提条件

- Extended Fast Software Upgrade 機能は Cisco IOS XE Amsterdam 17.3.2a 以降にサポートされています。アップグレードプロセスを開始する前に、スイッチにインストールされているソフトウェアのバージョンが Cisco IOS XE Amsterdam 17.3.2a 以降であることを確認します。
- **no boot manual** コマンドを使用して手動ブートを無効にする必要があります。

Extended Fast Software Upgrade の制約事項

次の制約事項は、スタンドアロンスイッチとスタック構成スイッチの両方に適用されます。

- この機能は、スイッチがインストールモードで動作している場合にのみサポートされます。
- この機能は、スパニングツリープロトコル（STP）のみが設定されたスイッチではサポートされません。さらに、スイッチには Rapid Spanning Tree Protocol（RSTP）または Multiple Spanning Tree Protocol（MSTP）を設定する必要があります。

- ルートデバイスでは、フォワーディングステートのデバイスポートのいずれかがSTPピア（STPが設定され、ルートデバイスに直接接続されているデバイス）に接続されている場合、Extended Fast Software Upgrade 機能はサポートされません。
- STP が設定され、ルートデバイスとして定義されていないデバイスの場合、Extended Fast Software Upgrade 機能は、STP ピアに接続されているフォワーディングステートのデバイスポートの数が 1 以下の場合にのみサポートされます。
- Extended Fast Software Upgrade の実行後、アプリケーション ホスティングが自動的に再起動しない場合があります。Cisco IOx の無効化と再有効化、アプリケーション ホスティングの設定、アプリケーションのインストール、アクティブ化、および再起動が必要になる場合があります。

スタック構成スイッチには、次の制約事項が適用されます。

- スタック構成スイッチが部分リング状態で設定されている場合、この機能はサポートされません。
- この機能は、 Bidirectional Forwarding Detection（BFD）が設定されているスタック構成スイッチではサポートされません。
- この機能は、MACsec Key Agreement（MKA）が設定されているスタック構成スイッチではサポートされません。
- この機能は、Cisco TrustSec が設定されているスタック構成スイッチではサポートされません。
- 設定された単方向リンク検出（UDLD）メッセージ間隔は、トラフィックのダウンタイム中は無視されます。間隔は、Extended Fast Software Upgrade の完了後に設定された値に復元されます。

Extended Fast Software Upgrade に関する情報

Extended Fast Software Upgrade により、ソフトウェアのリロードまたはアップグレード操作中のトラフィックのダウンタイムが削減されます。Fast Software Upgrade と比較して、スイッチの設定によっては、トラフィックのダウンタイムが 30 秒未満に短縮されます。Extended Fast Software Upgrade は、グレースフルリスタート機能（Cisco NSF とも呼ばれます）を使用して、ソフトウェアのアップグレードまたはリロード中に特定のルーティングプロトコルなどのスイッチ設定が影響を受けないようにします。

（注） Extended Fast Software Upgrade は、アクセスレイヤスイッチでのみ動作します。

Perpetual Power over Ethernet（PoE）が設定されている場合、ソフトウェアのリロードまたはアップグレード中に、接続されたデバイスに中断なく電力が供給されます。

Extended Fast Software Upgrade でサポートされるプロトコル

Extended Fast Software Upgrade 機能では、次のプロトコルがサポートされています。

（注） 次のプロトコル以外のすべてのプロトコルで、トラフィックのダウンタイムは 30 秒より長くなります。

- BGP（IPv4 および IPv6 アドレスファミリ）
- Flexible NetFlow
- IEEE 802.1X ポートベースの認証
- Intermediate System-to-Intermediate System（IS-IS）
- インターネット グループ管理プロトコル（IGMP）スヌーピング
- レイヤ 2 スイッチング
- リンク集約制御プロトコル（LACP）
- MAC 認証バイパス
- マルチキャストリスナー検出（MLD）スヌーピング
- Open Shortest Path First（OSPF）または OSPFv2 または OSPFv3
- Per VLAN Spanning Tree（PVST）
- QoS
- スタティックポートチャネル（モードオン）
- RSTP または MSTP を使用した STP
- UDLD
- VPN ルーティングおよび転送（VRF）
- Web 認証

スタンドアロンスイッチでの Extended Fast Software Upgrade の実行方法

次のセクションでは、スタンドアロンスイッチで Extended Fast Software Upgrade を実行する方法について説明します。

スタンドアロンスイッチでのソフトウェアのアップグレード

スタンドアロンスイッチのソフトウェアをアップグレードするには、次の手順を実行します。

始める前に

- すべてのルーティングプロトコルが**UP**ステータスにあることを確認するには、特権EXECモードで **show graceful-reload** コマンドを使用します。
- 必要に応じて、新しいソフトウェア用にディスク領域を解放するには、特権EXECモードで **install remove inactive** 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 モードを有効にします。 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** *image***activate reloadfast commit** 例： `Device# install add file bootflash: cat9k_iosxe.17.03.02.SPA.bin activate reloadfast commit`	新しいソフトウェアイメージでスイッチをアップグレードします。 *image* キーワードには、ファイルの場所（TFTP、HTTP、フラッシュドライブ）とイメージ名を含めます。

IPv6 が設定されたスタンドアロンスイッチでのソフトウェアのアップグレード

スタンドアロンスイッチのソフトウェアをアップグレードするには、次の手順を実行します。

始める前に

- すべてのルーティングプロトコルが**UP**ステータスにあることを確認するには、特権EXECモードで **show graceful-reload** コマンドを使用します。
- 必要に応じて、新しいソフトウェア用にディスク領域を解放するには、特権EXECモードで **install remove inactive** 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：	特権 EXEC モードを有効にします。

	コマンドまたはアクション	目的
	`Device> enable`	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	**ipv6 nd reachable-time** *seconds* 例： `Device(config)# ipv6 nd reachable-time 3600000`	到達可能性確認イベントの発生後、リモート IPv6 ノードが到達可能と判断されるまでの期限を設定します。
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	**write memory** 例： `Device# write memory`	スイッチの設定を保存します。
ステップ 6	**install add file** *image***activate reloadfast commit** 例： `Device# install add file bootflash: cat9k_iosxe.17.03.02.SPA.bin activate reloadfast commit`	新しいソフトウェアイメージでスイッチをアップグレードします。 *image* キーワードには、ファイルの場所（TFTP、HTTP、フラッシュドライブ）とイメージ名を含めます。

IPv6 MLD が設定されたスタンドアロンスイッチでのソフトウェアのアップグレード

IPv6 MLD が設定されたスタンドアロンスイッチのソフトウェアをアップグレードするには、次の手順を実行します。

始める前に

- すべてのルーティングプロトコルが**UP**ステータスにあることを確認するには、特権EXECモードで **show graceful-reload** コマンドを使用します。
- 必要に応じて、新しいソフトウェア用にディスク領域を解放するには、特権EXECモードで **install remove inactive** 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 モードを有効にします。 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	**ipv6 mld snooping** 例： `Device(config)# ipv6 mld snooping`	IPv6 MLD スヌーピングをイネーブルにします。
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	**write memory** 例： `Device# write memory`	スイッチの設定を保存します。
ステップ 6	**install add file** *image***activate reloadfast commit** 例： `Device# install add file bootflash: cat9k_iosxe.17.03.02.SPA.bin activate reloadfast commit`	新しいソフトウェアイメージでスイッチをアップグレードします。 *image* キーワードには、ファイルの場所（TFTP、HTTP、フラッシュドライブ）とイメージ名を含めます。

BGPが設定されたスタンドアロンスイッチでのソフトウェアのアップグレード

BGPが設定されたスタンドアロンスイッチのソフトウェアをアップグレードするには、次の手順を実行します。

始める前に

- すべてのルーティングプロトコルが**UP**ステータスにあることを確認するには、特権EXECモードで **show graceful-reload** コマンドを使用します。
- 必要に応じて、新しいソフトウェア用にディスク領域を解放するには、特権EXECモードで **install remove inactive** 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 モードを有効にします。 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	**router bgp** *autonomous-system-number* 例： `Device(config)# router bgp 65000`	BGP ルータ コンフィギュレーション モードを開始します。
ステップ 4	**bgp graceful-restart** 例： `Device(config-router)# bgp graceful-restart`	NSF 認識をスイッチで有効にします。NSF 認識はデフォルトでは無効です。
ステップ 5	**end** 例： `Device(config-router)# end`	特権 EXEC モードに戻ります。
ステップ 6	**write memory** 例： `Device# write memory`	スイッチの設定を保存します。
ステップ 7	**install add file** *image***activate reloadfast commit** 例： `Device# install add file bootflash: cat9k_iosxe.17.03.02.SPA.bin activate reloadfast commit`	新しいソフトウェアイメージでスイッチをアップグレードします。 *image* キーワードには、ファイルの場所（TFTP、HTTP、フラッシュドライブ）とイメージ名を含めます。

OSPFv3が設定されたスタンドアロンスイッチでのソフトウェアのアップグレード

OSPFv3 が設定されたスタンドアロンスイッチのソフトウェアをアップグレードするには、次の手順を実行します。

始める前に

- すべてのルーティングプロトコルが**UP**ステータスにあることを確認するには、特権EXECモードで **show graceful-reload** コマンドを使用します。
- 必要に応じて、新しいソフトウェア用にディスク領域を解放するには、特権EXECモードで **install remove inactive** 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 モードを有効にします。 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	**ipv6 nd reachable-time** *seconds* 例： `Device(config)# ipv6 nd reachable-time 3600000`	到達可能性確認イベントの発生後、リモート IPv6 ノードが到達可能と判断されるまでの期限を設定します。
ステップ 4	**snmp ifmib ifindex persist** 例： `Device(config)# snmp ifmib ifindex persist`	SNMP ifIndex の持続性をグローバルにイネーブルにします。
ステップ 5	**router ospfv3** *process-id* 例： `Device(config)# router ospfv3 1`	OSPFv3 ルータ コンフィギュレーション モードを開始します。
ステップ 6	**router-id** *ip-address* 例： `Device(config-router)# router-id 192.0.2.5`	OSPFv3 インスタンスの固定ルータ ID を設定します。
ステップ 7	**interface-id snmp-if-index** 例： `Device(config-router)# interface-id snmp-if-index`	特定のインターフェイスで SNMP ifIndex の持続性をイネーブルにします。
ステップ 8	**end** 例： `Device(config-router)# 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 9	**write memory** 例： `Device# write memory`	スイッチの設定を保存します。
ステップ 10	**install add file** *image***activate reloadfast commit** 例： `Device# install add file bootflash: cat9k_iosxe.17.03.02.SPA.bin activate reloadfast commit`	新しいソフトウェアイメージでスイッチをアップグレードします。 *image* キーワードには、ファイルの場所（TFTP、HTTP、フラッシュドライブ）とイメージ名を含めます。

スタンドアロンスイッチでのソフトウェアのリロード

スタンドアロンスイッチでソフトウェアをリロードするには、次の手順を実行します。

始める前に

すべてのルーティングプロトコルが**UP**ステータスにあることを確認するには、特権EXECモードで **show graceful-reload** 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 モードを有効にします。 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	**reload fast** 例： `Device# reload fast`	ソフトウェアをアップグレードせずにスイッチをリロードします。

BGPが設定されたスタンドアロンスイッチでのソフトウェアのリロード

BGPが設定されたスタンドアロンスイッチでソフトウェアをリロードするには、次の手順を実行します。

始める前に

すべてのルーティングプロトコルが**UP**ステータスにあることを確認するには、特権EXECモードで **show graceful-reload** 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 モードを有効にします。 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	**router bgp** *autonomous-system-number* 例： `Device(config)# router bgp 65000`	BGP ルータ コンフィギュレーション モードを開始します。
ステップ 4	**bgp graceful-restart** 例： `Device(config-router)# bgp graceful-restart`	NSF 認識をスイッチで有効にします。NSF 認識はデフォルトでは無効です。
ステップ 5	**end** 例： `Device(config-router)# end`	特権 EXEC モードに戻ります。
ステップ 6	**write memory** 例： `Device# write memory`	スイッチの設定を保存します。
ステップ 7	**reload fast** 例： `Device# reload fast`	ソフトウェアをアップグレードせずにスイッチをリロードします。

OSPFv3 が設定されたスタンドアロンスイッチでのソフトウェアのリロード

OSPv3が設定されたスタンドアロンスイッチでソフトウェアをリロードするには、次の手順を実行します。

始める前に

すべてのルーティングプロトコルが**UP**ステータスにあることを確認するには、特権EXECモードで **show graceful-reload** 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 モードを有効にします。 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	**ipv6 nd reachable-time** *seconds* 例： `Device(config)# ipv6 nd reachable-time 3600000`	到達可能性確認イベントの発生後、リモート IPv6 ノードが到達可能と判断されるまでの期限を設定します。
ステップ 4	**snmp ifmib ifindex persist** 例： `Device(config)# snmp ifmib ifindex persist`	SNMP ifIndex の持続性をグローバルにイネーブルにします。
ステップ 5	**router ospfv3** *process-id* 例： `Device(config)# router ospfv3 1`	OSPFv3 ルータ コンフィギュレーション モードを開始します。
ステップ 6	**router-id** *ip-address* 例： `Device(config-router)# router-id 192.0.2.5`	OSPFv3 インスタンスの固定ルータ ID を設定します。
ステップ 7	**interface-id snmp-if-index** 例： `Device(config-router)# interface-id snmp-if-index`	特定のインターフェイスで SNMP ifIndex の持続性をイネーブルにします。
ステップ 8	**end** 例： `Device(config-router)# end`	特権 EXEC モードに戻ります。
ステップ 9	**write memory** 例： `Device# write memory`	スイッチの設定を保存します。
ステップ 10	**reload fast** 例： `Device# reload fast`	ソフトウェアをアップグレードせずにスイッチをリロードします。

スタック構成スイッチでの Extended Fast Software Upgrade の実行方法

ここでは、スタック構成のスイッチで Extended Fast Software Upgrade を実行する方法について説明します。

スタック構成スイッチでのソフトウェアのアップグレード

スタック構成スイッチでソフトウェアをアップグレードするには、次の手順を実行します。

始める前に

必要に応じて、新しいソフトウェア用にディスク領域を解放するには、特権 EXEC モードで **install remove inactive** 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 モードを有効にします。 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** *image***activate reloadfast commit** 例： `Device# install add file bootflash: cat9k_iosxe.17.03.02.SPA.bin activate reloadfast commit`	次のプロセスが発生します。 1. アクティブ、スタンバイ、およびメンバースイッチのイメージをアップグレードします。 2. スタンバイスイッチとメンバースイッチを再起動します。アクティブスイッチを再起動し、スイッチの切り替えが行われます。スタンバイスイッチがアクティブスイッチになり、アクティブスイッチがスタンバイスイッチになります。 *image* キーワードには、ファイルの場所（TFTP、HTTP、フラッシュドライブ）とイメージ名を含めます。

BGPが設定されたスタック構成スイッチでのソフトウェアのアップグレード

BGPが設定されたスタック構成スイッチのソフトウェアをアップグレードするには、次の手順を実行します。

始める前に

- すべてのルーティングプロトコルが**UP**ステータスにあることを確認するには、特権EXECモードで **show graceful-reload** コマンドを使用します。
- 必要に応じて、新しいソフトウェア用にディスク領域を解放するには、特権EXECモードで **install remove inactive** 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 モードを有効にします。 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	**router bgp** *autonomous-system-number* 例： `Device(config)# router bgp 65000`	BGP ルータ コンフィギュレーション モードを開始します。
ステップ4	**bgp graceful-restart all** 例： `Device(config-router)# bgp graceful-restart`	NSF 認識をスイッチで有効にします。NSF 認識はデフォルトでは無効です。
ステップ5	**end** 例： `Device(config-router)# end`	特権 EXEC モードに戻ります。
ステップ6	**write memory** 例： `Device# write memory`	スイッチの設定を保存します。

	コマンドまたはアクション	目的
ステップ 7	**install add file** *image***activate reloadfast commit** 例： `Device# install add file bootflash: cat9k_iosxe.17.03.02.SPA.bin activate reloadfast commit`	新しいソフトウェアイメージでスイッチをアップグレードします。 *image* キーワードには、ファイルの場所（TFTP、HTTP、フラッシュドライブ）とイメージ名を含めます。

IS-ISが設定されたスタック構成スイッチでのソフトウェアのアップグレード

IS-IS が設定されたスタック構成スイッチのソフトウェアをアップグレードするには、次の手順を実行します。

始める前に

必要に応じて、新しいソフトウェア用にディスク領域を解放するには、特権 EXEC モードで **install remove inactive** 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 モードを有効にします。 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	**router isis** *area-tag* 例： `Device(config)# router isis tag1`	IS-IS ルーティングプロトコルをイネーブルにして IS-IS プロセスを指定し、スイッチをルータ コンフィギュレーション モードにします。
ステップ 4	**nsf** {**cisco** | **ietf**} 例： `Device(config-router)# nsf cisco` OR `Device(config-router)# nsf ietf`	IS-IS 用 NSF をイネーブルにします。 • **ietf**：IETF ドラフトベースの再起動をサポートするネットワーキングスイッチとの隣接関係がサポートしている同種ネットワークで IS-IS をイネーブルにする。

	コマンドまたはアクション	目的
		• **cisco**：NSF 認識ネットワーキングスイッチとの隣接関係がない同種ネットワークで IS-IS を実行する。
ステップ 5	**end** 例： `Device(config-router)# end`	特権 EXEC モードに戻ります。
ステップ 6	**write memory** 例： `Device# write memory`	スイッチの設定を保存します。
ステップ 7	**install add file** *image***activate reloadfast commit** 例： `Device# install add file bootflash: cat9k_iosxe.17.03.02.SPA.bin activate reloadfast commit`	新しいソフトウェアイメージでスイッチをアップグレードします。 *image* キーワードには、ファイルの場所（TFTP、HTTP、フラッシュドライブ）とイメージ名を含めます。

スタック構成スイッチでのソフトウェアのリロード

スタック構成スイッチでソフトウェアをリロードするには、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 モードを有効にします。 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	**reload fast** 例： `Device# reload fast`	次のプロセスが発生します。 1. スタンバイスイッチとメンバースイッチをリロードします。 2. アクティブスイッチをリロードし、スイッチの切り替えが行われます。スタンバイスイッチがアクティブスイッチになり、アクティブスイッチが新しいスタンバイスイッチになります。

BGPが設定されたスタック構成スイッチでのソフトウェアのリロード

BGPが設定されたスタック構成スイッチでソフトウェアをリロードするには、次の手順を実行します。

始める前に

すべてのルーティングプロトコルが**UP**ステータスにあることを確認するには、特権EXECモードで **show graceful-reload** 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 モードを有効にします。 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	**router bgp** *autonomous-system-number* 例： `Device(config)# router bgp 65000`	BGP ルータ コンフィギュレーション モードを開始します。
ステップ 4	**bgp graceful-restart all** 例： `Device(config-router)# bgp graceful-restart all`	スタック内のすべてのスイッチで NSF 認識をイネーブルにします。NSF 認識はデフォルトでは無効です。
ステップ 5	**end** 例： `Device(config-router)# end`	特権 EXEC モードに戻ります。
ステップ 6	**write memory** 例： `Device# write memory`	スイッチの設定を保存します。
ステップ 7	**reload fast** 例： `Device# reload fast`	ソフトウェアをアップグレードせずにスイッチをリロードします。

IS-ISが設定されたスタック構成スイッチでのソフトウェアのリロード

IS-IS が設定されたスタック構成スイッチでソフトウェアをリロードするには、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 モードを有効にします。 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	**router isis** *area-tag* 例： `Device(config)# router isis tag1`	IS-IS ルーティングプロトコルをイネーブルにして IS-IS プロセスを指定し、スイッチをルータ コンフィギュレーション モードにします。
ステップ 4	**nsf** {**cisco** | **ietf**} 例： `Device(config-router)# nsf cisco` OR `Device(config-router)# nsf ietf`	IS-IS 用 NSF をイネーブルにします。 • **ietf**：IETF ドラフトベースの再起動をサポートするネットワーキングスイッチとの隣接関係がサポートしている同種ネットワークで IS-IS をイネーブルにする。 • **cisco**：NSF 認識ネットワーキングスイッチとの隣接関係がない同種ネットワークで IS-IS を実行する。
ステップ 5	**end** 例： `Device(config-router)# end`	特権 EXEC モードに戻ります。
ステップ 6	**write memory** 例： `Device# write memory`	スイッチの設定を保存します。
ステップ 7	**reload fast** 例： `Device# reload fast`	ソフトウェアをアップグレードせずにスイッチをリロードします。

ソフトウェアのアップグレードまたはリロードの確認

ソフトウェアのアップグレードまたはリロードが成功したことを確認するには、特権 EXEC モードで次のコマンドを使用します。

表 *24 :* ソフトウェアのアップグレードまたはリロードを確認するコマンド

コマンド	目的
show version	デバイスのハードウェアおよびソフトウェア情報を表示します。
show log | in FAST	Extended Fast Software Upgrade を使用したソフトウェアアップグレードまたはリロードが完了したかどうかを表示します。
show install summary	アクティブなパッケージに関する情報を表示します。
show install log	インストール要求に関する情報を表示します。
show version running	現在実行中のファイルに関する情報を表示します。
show version | in reason	最後のリロードの理由を表示します。

その他の参考資料

関連資料

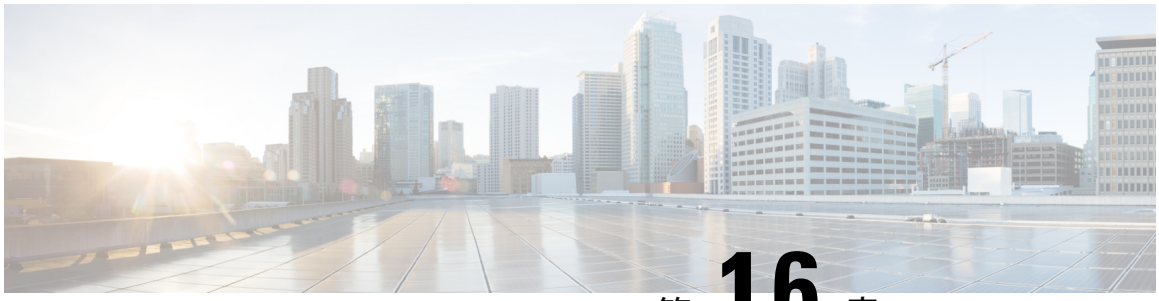
関連項目	マニュアル タイトル
ルーティングプロトコルに関する情報	*Software Configuration Guide (Catalyst 9300 Switches)*の「*IP Routing Configuration Guide*」を参照してください。
STP、PVST、およびUDLDに関する情報	*Software Configuration Guide (Catalyst 9300 Switches)*の『レイヤ *2* 設定ガイド』を参照してください。
無停止型 PoE に関する情報	*Software Configuration Guide (Catalyst 9300 Switches)*の「*Network Powered Lighting Configuration Guide*」を参照してください。
アプリケーション ホスティングに関する情報	*Software Configuration Guide (Catalyst 9300 Switches)*の「*Programmability Configuration Guide*」を参照してください。

Extended Fast Software Upgrade の機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

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

リリース	機能	機能情報
Cisco IOS XE Amsterdam 17.3.2a	Extended Fast Software Upgrade	Extended Fast Software Upgrade により、ソフトウェアのリロードまたはアップグレード操作中のトラフィックのダウンタイムが削減されます。 この機能は、Cisco Catalyst 9300 シリーズ スイッチの 9300 および 9300L スイッチモデルでのみサポートされるようになりました。



第 16 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

- SMU は、インストールモードを使用したパッチのみをサポートします。

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

SMU は、システムにインストールして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ます。

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 Cisco 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。

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。

SMU をインストールするには、次の基本的な手順を実行します。

1. ファイルシステムに SMU を追加します。
2. システムで SMU をアクティブ化します。
3. リロードが繰り返されても持続させるための SMU の変更をコミットします。

SMU のワークフロー

SMU プロセスは、シスコカスタマーサポート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 [Cisco Software Download]https://www.cisco.com/c/en_in/support/index.html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、パッケージの内容を記述するいくつかのメタデータ、および SMU が要求されている報告済みの問題の修正とともに、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 パッケージは、公開キーインフラストラクチャ（PKI）コンポーネントのパッチ適用もサポートします。

SMU のリロード

SMU タイプは、インストールされている SMU が対応するシステムに与える影響を示します。SMU がトラフィックに影響を与えない場合や、SMU によってデバイスの再起動、リロード、またはスイッチオーバーが発生する場合があります。リロードが必要かどうかを確認するには、**show install package flash**: *filename* コマンドを実行します。

ホットパッチを使用すると、SMU はアクティブ化後に有効になり、システムをリロードする必要がありません。SMU がコミットされると、リロードが繰り返されても変更が持続します。場合によっては、SMU でオペレーティングシステムのコールド（完全）リロードが必要になることがあります。このアクションは、リロードの間、トラフィックフローに影響します。コールドリロードが必要な場合、ユーザーにはアクションを確認するプロンプトが表示されます。

ソフトウェア メンテナンスの更新の管理方法

ここでは、SMU の管理に関する情報について説明します。

単一のコマンドまたは個別のコマンドを使用して SMU パッケージのインストール、アクティブ化、コミットを行うことができます。

SMU パッケージのインストール

このタスクでは、SMU パッケージをインストールするための **install add file activate commit** コマンドの使用方法を示します。

手順

	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 モードを有効にします。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 flash:** *filename* [**activate commit**] 例： `Device# install add file flash:cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin activate commit`	メンテナンス更新パッケージをフラッシュからコピーし、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、SMU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。 また、リモートロケーションから（FTP、HTTP、HTTPS、または TFTP を使用して）メンテナンス更新パッケージをコピーすることもできます。 （注） TFTP を使用して SMU ファイルをコピーする場合は、ブートフラッシュを使用して SMU をアクティブにします。
ステップ 3	**exit** 例： `Device# exit`	特権 EXEC モードを終了し、ユーザー EXEC モードに戻ります。

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 モードを有効にします。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 flash:** *filename* 例： `Device# install add file flash:cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin`	SMU パッケージをソースの場所からデバイスにコピーし（ソースの場所がリモートの場合）、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、必要に応じてすべてのメンバノードまたは FRU に SMU パッケージを追加します。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も確認します。また、`package/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 flash:** *filename* 例： `Device# install activate add file flash:cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.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。
ステップ 4	**install commit** 例： `Device# install commit`	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、システムの起動時、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。

	コマンドまたはアクション	目的
ステップ 6	**install deactivate file flash:** *filename* 例： `Device# install deactivate file flash:cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin`	アクティブパッケージを非アクティブ化し、パッケージのステータスを更新します。
ステップ 7	**install remove** {**file flash:** *filename* | **inactive**} 例： `Device# install remove file flash:cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin`	指定した SMU が非アクティブかどうかを確認し、非アクティブの場合はファイルシステムから削除します。**inactive** オプションは、非アクティブなパッケージをファイルシステムからすべて削除します。
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`	パッケージのインストールステータス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

次に、SMU の設定例を示します。

例：SMU の管理

（注）
- このセクションでは、ホットパッチ SMU の例を使用しています。

次に、SMU ファイルをフラッシュにコピーする例を示します。

```
Device# copy ftp://172.16.0.10//auto/ftpboot/user/
cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin

flash:
Destination filename
[cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin]?
Accessing ftp://172.16.0.10//auto/ftpboot/folder1/
cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin...
Loading /auto/ftpboot/folder1/
cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin from
172.16.0.10 (via GigabitEthernet0): !
```

```
[OK - 17668 bytes]
17668 bytes copied in 0.058 secs (304621 bytes/sec)
```

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
flash:cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin

install_add: START Mon Mar  5 21:48:51 PST 2018
install_add: Adding SMU

--- Starting initial file syncing ---
Info: Finished copying
flash:cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin to
the selected switch(es)
Finished initial file syncing

Executing pre scripts....

Executing pre scripts done.
--- Starting SMU Add operation ---
Performing SMU_ADD on all members
  [1] SMU_ADD package(s) on switch 1
  [1] Finished SMU_ADD on switch 1
Checking status of SMU_ADD on [1]
SMU_ADD: Passed on [1]
Finished SMU Add operation

SUCCESS: install_add
/flash/cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin Mon
 Mar  5 21:49:00 PST 2018
```

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

```
Device# show install summary

[ Switch 1 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
SMU   I
flash:cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin
IMG   C    16.9.1.0.43131

--------------------------------------------------------------------------------
Auto abort timer: inactive
--------------------------------------------------------------------------------
```

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
flash:cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin

install_activate: START Mon Mar  5 21:49:22 PST 2018
install_activate: Activating SMU
Executing pre scripts....

Executing pre scripts done.
```

```
--- Starting SMU Activate operation ---
Performing SMU_ACTIVATE on all members
  [1] SMU_ACTIVATE package(s) on switch 1
  [1] Finished SMU_ACTIVATE on switch 1
Checking status of SMU_ACTIVATE on [1]
SMU_ACTIVATE: Passed on [1]
Finished SMU Activate operation

SUCCESS: install_activate
/flash/cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin Mon
 Mar  5 21:49:34 PST 2018
```

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_LATEST_20180302_085005_2 - SMU-PATCHED
Cisco IOS Software [Fuji], Catalyst L3 Switch Software (CAT9K_IOSXE), Experimental Version
 16.9.20180302:
085957 [polaris_dev-/nobackup/mcpre/BLD-BLD_POLARIS_DEV_LATEST_20180302_085005 166]
Copyright (c) 1986-2018 by Cisco Systems, Inc.
Compiled Fri 02-Mar-18 09:50 by mcpre
...
```

次に示すのは、**show install summary** コマンドが SMU 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

[ Switch 1 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
SMU   U
flash:cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin
IMG   C    16.9.1.0.43131

--------------------------------------------------------------------------------
Auto abort timer: active on install_activate, time before rollback - 01:59:50
--------------------------------------------------------------------------------
```

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

```
Device# show install active

[ Switch 1 ] Active Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
SMU   U
flash:cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin
IMG   C    16.9.1.0.43131
```

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

```
Device# install commit
```

```
install_commit: START Mon Mar  5 21:50:52 PST 2018
install_commit: Committing SMU
Executing pre scripts....

Executing pre scripts done.
--- Starting SMU Commit operation ---
Performing SMU_COMMIT on all members
  [1] SMU_COMMIT package(s) on switch 1
  [1] Finished SMU_COMMIT on switch 1
Checking status of SMU_COMMIT on [1]
SMU_COMMIT: Passed on [1]
Finished SMU Commit operation

SUCCESS: install_commit
/flash/cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin Mon
 Mar  5 21:51:01 PST 2018
```

次に示すのは、**show install summary** コマンドが、更新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

[ Switch 1 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
SMU   C
flash:cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin
IMG   C    16.9.1.0.43131

--------------------------------------------------------------------------------
Auto abort timer: inactive
--------------------------------------------------------------------------------
```

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 committed

install_rollback: START Mon Mar  5 21:52:18 PST 2018
install_rollback: Rolling back SMU
Executing pre scripts....

Executing pre scripts done.

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

SUCCESS: install_rollback
/flash/cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin Mon
 Mar  5 21:52:30 PST 2018
```

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

```
Device# show install summary

[ Switch 1 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
IMG   C    16.9.1.0.43131

--------------------------------------------------------------------------------
Auto abort timer: inactive
--------------------------------------------------------------------------------
```

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
flash:cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin

install_deactivate: START Mon Mar  5 21:54:06 PST 2018
install_deactivate: Deactivating SMU
Executing pre scripts....

Executing pre scripts done.

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

SUCCESS: install_deactivate
/flash/cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin Mon
 Mar  5 21:54:17 PST 2018
```

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

```
Device# show install summary

[ Switch 1 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
SMU   D
flash:cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin
IMG   C    16.9.1.0.43131

--------------------------------------------------------------------------------
Auto abort timer: active on install_deactivate, time before rollback - 01:59:50
--------------------------------------------------------------------------------
```

次に、デバイスから SMU を削除する例を示します。

```
Device# install remove file
flash:cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin

install_remove: START Mon Mar  5 22:03:50 PST 2018
install_remove: Removing SMU
```

```
Executing pre scripts....

Executing pre scripts done.

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

SUCCESS: install_remove
/flash/cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin Mon
 Mar  5 22:03:58 PST 2018
```

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

```
Device# show install summary

[ Switch 1 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
IMG   C    16.9.1.0.43131

--------------------------------------------------------------------------------
Auto abort timer: inactive
--------------------------------------------------------------------------------
```

ソフトウェアメンテナンスアップグレードのその他の参考資料

関連資料

関連項目	マニュアル タイトル
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 9300* シリーズ スイッチ）

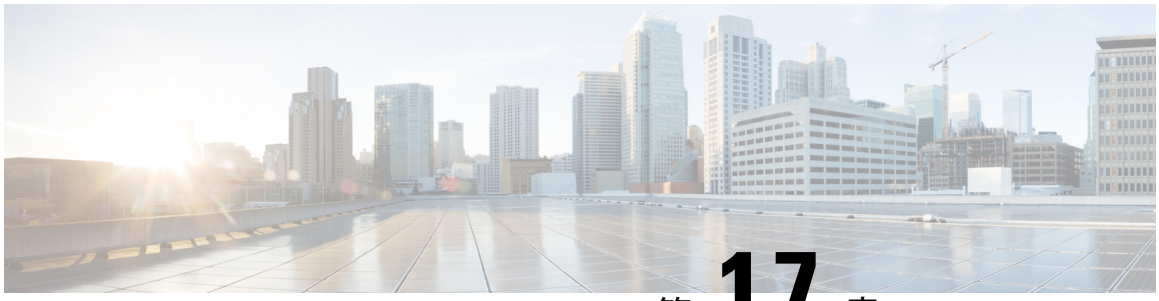
ソフトウェアメンテナンスアップグレードの機能の履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

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

リリース	機能	機能情報
Cisco IOS XE Everest 16.6.1	ソフトウェア メンテナンス アップグレード（SMU）	SMUは、システムにインストールして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ができるパッケージです。
Cisco IOS XE Fuji 16.9.1	ホットパッチ	ホットパッチを使用すると、SMU はアクティブ化後に有効になり、システムをリロードする必要がありません。
Cisco IOS XE Gibraltar 16.10.1	Public Key Infrastructure（PKI）	SMU パッケージは、PKI コンポーネントのパッチ適用をサポートします。

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



第 17 章

フラッシュ ファイル システムの操作

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またはスタックメンバから見ると、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人のユーザーのみが、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）。

```
Device# show file systems
Size(b) Free(b) Type Flags Prefixes
- - opaque rw system:
```

```
- - opaque rw tmpsys:
1651314688 1559785472 disk rw crashinfo:
* 11353194496 9693396992 disk rw flash:
8049967104 7959392256 disk ro webui:
- - opaque rw null:
- - opaque ro tar:
- - network rw tftp:
2097152 2080848 nvram rw nvram:
- - opaque wo syslog:
- - network rw rcp:
- - network rw http:
- - network rw ftp:
- - network rw scp:
- - network rw https:
- - opaque ro cns:
```

```
Device# show file systems
File Systems:
Size(b) Free(b) Type Flags Prefixes
- - opaque rw system:
- - opaque rw tmpsys:
1651314688 1467920384 disk rw crashinfo:
* 11353194496 6942072832 disk rw flash:
7723847680 7646384128 disk ro webui:
- - opaque rw null:
- - opaque ro tar:
- - network rw tftp:
2097152 2089932 nvram rw nvram:
- - network rw rcp:
- - network rw http:
- - network rw ftp:
- - network rw scp:
- - network rw https:
- - opaque ro cns:
118014062592 111933124608 disk rw usbflash1:
```

```
Device# show file systems
File Systems:

       Size(b)       Free(b)      Type  Flags  Prefixes
             -             -    opaque     rw   system:
             -             -    opaque     rw   tmpsys:
*  11250098176    9694093312      disk     rw   bootflash: flash:
    1651314688    1232220160      disk     rw   crashinfo:
  118148280320  112084115456      disk     rw   disk0:
     189628416     145387520      disk     rw   usbflash0:
    7763918848    7696850944      disk     ro   webui:
             -             -    opaque     rw   null:
             -             -    opaque     ro   tar:
             -             -   network     rw   tftp:
      33554432      33532852     nvram     rw   nvram:
             -             -    opaque     wo   syslog:
             -             -   network     rw   rcp:
             -             -   network     rw   http:
             -             -   network     rw   ftp:
             -             -   network     rw   scp:
```

```
-             -   network     rw   https:
-             -    opaque     ro   cns:
```

この例では、usbflash1 filesystem 形式を表示します。

```
Device#show usbflash1: filesys
Filesystem: usbflash1
Filesystem Path: /vol/usb1
Filesystem Type: ext4
Mounted: Read/Write
```

次の例では、デバイススタックを示します。この例では、アクティブなデバイスはスタックメンバ 1 です。スタックメンバ 2 のファイルシステムは flash-2: として、スタックメンバ 3 のファイルシステムは flash-3: として表示されるといった具合に、まで続きます。また、この例では、次のように、crashinfo ディレクトリと、アクティブなデバイスに接続された USB フラッシュドライブも示します。

```
Device# show file systems
File Systems:

Size(b) Free(b) Type Flags Prefixes
- - opaque rw system:
- - opaque rw tmpsys:
1651314688 1520742400 disk rw crashinfo: crashinfo-1:
1651507200 1516240896 disk rw crashinfo-2: stby-crashinfo:
1651507200 1517289472 disk rw crashinfo-3:
1651507200 1519386624 disk rw crashinfo-4:
1651507200 1524629504 disk rw crashinfo-5:
1651507200 1523580928 disk rw crashinfo-6:
1651507200 1517289472 disk rw crashinfo-7:
1651507200 1526726656 disk rw crashinfo-8:
* 11353194496 7916576768 disk rw flash: flash-1:
11353980928 7944011776 disk rw flash-2: stby-flash:
11353980928 7876902912 disk rw flash-3:
11353980928 7944011776 disk rw flash-4:
11353980928 7939817472 disk rw flash-5:
11353980928 7944011776 disk rw flash-6:
11353980928 7944011776 disk rw flash-7:
11353980928 7944011776 disk rw flash-8:
3824013312 3756507136 disk ro webui:
- - opaque rw null:
- - opaque ro tar:
- - network rw tftp:
2097152 2052489 nvram rw nvram:
- - opaque wo syslog:
- - network rw rcp:
- - network rw http:
- - network rw ftp:
- - network rw scp:
- - network rw https:
- - opaque ro cns:
2097152 2052489 nvram rw stby-nvram:
- - nvram rw stby-rcsf:
- - opaque rw revrcsf:
```

次の例では、デバイススタックを示します。この例では、アクティブなデバイスはスタックメンバ 2 です。スタックメンバ 1 のファイルシステムは flash-1: として、スタックメンバ 2 のファイルシステムは flash-2: として、スタックメンバ 3 のファイルシステムは flash-3: として表示さ

れるといった具合に、まで続きます。また、この例では、次のように、crashinfoディレクトリと、アクティブなデバイスに接続された USB フラッシュドライブも示します。

```
Device# show file systems
File Systems:

       Size(b)       Free(b)      Type  Flags  Prefixes
             -             -    opaque     rw   system:
             -             -    opaque     rw   tmpsys:
    1651314688    1565089792      disk     rw   crashinfo: crashinfo-2:
    1651507200    1560281088      disk     rw   crashinfo-1:
    1651507200    1562378240      disk     rw   crashinfo-3: stby-crashinfo:
*  11353194496   10735611904      disk     rw   flash: flash-2:
   11353980928   10152312832      disk     rw   flash-1:
   11353980928    2161115136      disk     rw   flash-3: stby-flash:
   15243046912   14423638016      disk     rw   usbflash0: usbflash0-2:
     520093696     520093696      disk     rw   usbflash0-1:
    3497074688    3417554944      disk     ro   webui:
             -             -    opaque     rw   null:
             -             -    opaque     ro   tar:
             -             -   network     rw   tftp:
       2097152       2085334     nvram     rw   nvram:
             -             -   network     rw   rcp:
             -             -   network     rw   http:
             -             -   network     rw   ftp:
             -             -   network     rw   scp:
             -             -   network     rw   https:
             -             -    opaque     ro   cns:
   21003628544   19867037696      disk     rw   usbflash1: usbflash1-2:
  118014083072  111933390848      disk     rw   usbflash1-3: stby-usbflash1:
       2097152       2085334     nvram     rw   stby-nvram:
             -             -     nvram     rw   stby-rcsf:
             -             -    opaque     rw   revrcsf:
```

表 ***25 : 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。

フィールド	値
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 サーバーや HTTP サーバー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。
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
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 サーバーです。 **http**：HTTP サーバーです。 **https**：セキュア HTTP サーバーです。 **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）サーバーです。 **scp**：Session Control Protocol（SCP）サーバーです。 **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 サーバーです。 **usbflash0**：USB フラッシュ メモリです。 **usbflash1:**：外部の 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 *26 :* 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、別のユーザー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。

```
Device# 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)
Device#
```

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 モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**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。 スタックのデバイスメンバのフラッシュ パーティションにアクセスするには、flash-*n* を使用します（*n* はスタックメンバ番号です）。例えば、flash-4。
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：	デフォルトディレクトリに移動します。

	コマンドまたはアクション	目的
	`Device# cd`	

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** *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 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。*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** コマンドは、現在実行中のコンフィギュレーション ファイルをフラッシュメモリの NVRAM セクションに保存し、システム初期化の際にコンフィギュレーションファイルとして使用されるようにします。

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:、scp:、http:、https: 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
- SCP：scp:[[//username [:password]@location]/directory]/filename
- HTTP：http:[[//username [:password]@location]/directory]/filename
- HTTPS：https:[[//username [:password]@location]/directory]/filename

（注） パスワードに特殊文字「@」を含めることはできません。文字「@」を使用すると、コピーでサーバーの IP アドレスを解析できません。

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に（たとえば、**copy flash: flash:** コマンドは無効）

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。**archive download-sw** コマンドでインストールされ、不要になった古いソフトウェアイメージを削除するには、**/force** キーワードおよび **/recursive** キーワードを使用します。

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**： `Device# 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： `Device# 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： `Device# 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： `Device# 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	マニュアル タイトル
flash: ファイル システムの管理コマンド	*Cisco IOS Configuration Fundamentals Command Reference*

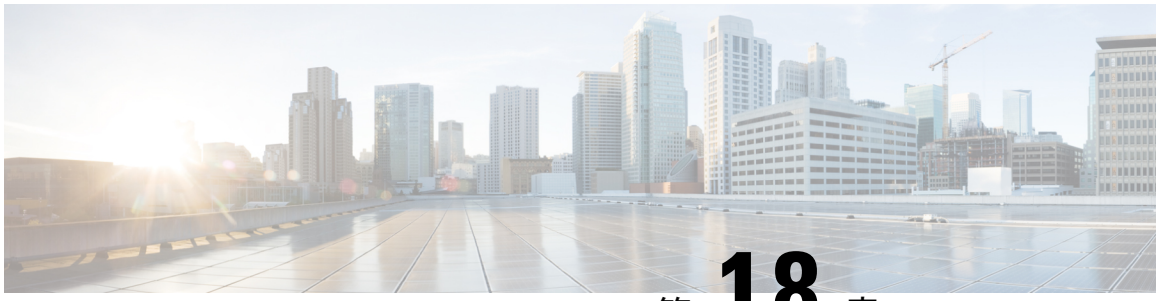
フラッシュファイルシステムの機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

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

リリース	機能	機能情報
Cisco IOS XE Everest 16.5.1a	フラッシュファイルシステム	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。

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



第 18 章

初期設定へのリセットの実行

初期設定へのリセット実行の前提条件

- 初期設定へのリセットプロセスを開始する前に、現在のイメージ、設定、および個人データを含むすべてのソフトウェアイメージがバックアップされていることを確認します。
- 初期設定へのリセットプロセスが進行中の場合は、電源の中断がないことを確認します。
- 初期設定へのリセットプロセスを開始する前に、In-Service Software Upgrade（ISSU）または In-Service Software Downgrade（ISSD）が進行中でないことを確認します。

初期設定へのリセット実行の制限事項

- ソフトウェアパッチは、デバイスにインストールされている場合、初期設定へのリセットプロセス後に復元されません。
- VTY セッションを介して **factory-reset** コマンドが発行された場合、初期設定へのリセットプロセスの完了後にセッションは復元されません。

初期設定へのリセットの実行に関する情報

初期設定にリセットすると、デバイスに保存されているお客様固有のデータがすべて消去され、デバイスの設定は出荷時の元の設定に復元されます。消去されるデータには、設定、ログファイル、ブート変数、コアファイル、および連邦情報処理標準関連（FIPS 関連）のキーなどのクレデンシャルが含まれます。NIST SP 800-88 Rev. 1 で説明されているように、消去は clear メソッドと一致します。

初期設定へのリセットプロセスは、次のシナリオで使用されます。

- デバイスの返品許可（RMA）：RMA のためにデバイスをシスコに返送する必要がある場合は、そのデバイスの RMA 証明書を取得する前に、お客様固有のデータをすべて削除してください。
- 侵害を受けたデバイスのリカバリ：デバイスに保存されているキーマテリアルまたはクレデンシャルが侵害を受けた場合は、デバイスを初期設定にリセットし、デバイスを再設定してください。

初期設定へのリセット時、デバイスはリロードされ、ROMMON モードを開始します。初期設定へのリセット後、デバイスは、ソフトウェアの検索とロードに必要な **MAC_ADDRESS** 変数と **SERIAL_NUMBER** 変数を含むすべての環境変数を削除します。ROMmon モードでリセットを実行すると、環境変数は自動的に設定されます。BAUD rate 環境変数は、初期設定へのリセット後にデフォルト値に戻ります。BAUD rate と console speed が常に同じであることを確認してください。同じでない場合、コンソールは応答しなくなります。

ROMmon モードでのシステムリセットが完了したら、USB または TFTP を使用して Cisco IOS イメージを追加します。

次の表に、初期設定へのリセットプロセス中に消去および保持されるデータの詳細を示します。

表 *27*：初期設定へのリセット時に消去および保持されるデータ

消去されるデータ	保持されるデータ
現在のブートイメージを含むすべての Cisco IOS イメージ	リモート Field-Replaceable Unit（FRU）からのデータ
クラッシュ情報とログ	コンフィギュレーション レジスタの値
ユーザーデータ、スタートアップおよび実行コンフィギュレーション、および Serial Advanced Technology Attachment（SATA）、SSD、USB などのリムーバブル ストレージ デバイスの内容	—

消去されるデータ	保持されるデータ
FIPS 関連キーなどのクレデンシャル	セキュアな固有デバイス識別子（SUDI）証明書、公開キーインフラストラクチャ（PKI）キーなどのクレデンシャル
オンボード障害ロギング（OBFL）ログ	ライセンス
ユーザーが追加した ROMmon 変数	—

初期設定へのリセットの実行方法

初期設定へのリセットを実行するには、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 モードを有効にします。 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	• スタンドアロンデバイスの場合： **factory-reset** {**all** ［**secure 3-pass**］| **config** ｜**boot-vars**} • スタック構成のデバイスの場合： **factory-reset** {**all** ［**secure 3-pass**］| **config** ｜**boot-vars**｜**switch** {*switch-number* ｜**all**{**all** ［**secure 3-pass**］| **config** ｜**boot-vars**}} 例： `Device# factory-reset all` または `Device# factory-reset switch 1 all config`	デバイスを出荷時の設定にリセットします。 **factory reset** コマンドを使用するために必要なシステム設定はありません。 次のオプションを使用できます。 • **all**：NVRAM のすべての内容、現在のブートイメージ、ブート変数、起動コンフィギュレーションと実行コンフィギュレーションのデータ、およびユーザーデータを含むすべての Cisco IOS イメージを消去します。このオプションを使用することを推奨します。 • **secure 3-pass**：3-pass 上書きでデバイスからすべての内容を消去します。 • Pass 1：すべてのアドレス可能な場所を 2 進数のゼロで上書きします。

	コマンドまたはアクション	目的
		• Pass 2：すべてのアドレス可能な場所を2進数の1で上書きします。 • Pass 3：すべてのアドレス可能な場所をランダムビットパターンで上書きします。 （注） このオプションは、他のオプションの実行にかかる時間の約 3 倍の時間がかかります。 • **config**：スタートアップ コンフィギュレーションをリセットします。 • **boot-vars**：ユーザーによって追加されたブート変数を消去します。 • **switch** {*switch-number* | **all**}: • *switch-number*：スイッチ番号を指定します。指定できる範囲は 1 ～ 16 です。 • **all**：スタック内のすべてのスイッチを選択します。 初期設定へのリセットプロセスが正常に完了すると、デバイスがリブートして ROMmon モードになります。

初期設定へのリセットを実行するための設定例

次に、スタンドアロンスイッチで初期設定へのリセットを実行する例を示します。

```
Device> enable
Device# factory-reset all

The factory reset operation is irreversible for all operations. Are you sure? [confirm]
The following will be deleted as a part of factory reset:
1: Crash info and logs
2: User data, startup and running configuration
3: All IOS images, including the current boot image
4: OBFL logs
5: User added rommon variables
6: Data on Field Replaceable Units(USB/SSD/SATA)
```

```
The system will reload to perform factory reset.
It will take some time to complete and bring it to rommon.
You will need to load IOS image using USB/TFTP from rommon after
this operation is completed.
DO NOT UNPLUG THE POWER OR INTERRUPT THE OPERATION
Are you sure you want to continue? [confirm]
```

次に、スタック構成デバイスで初期設定へのリセットを実行する例を示します。

```
Device> enable
Device# factory-reset switch all all
The factory reset operation is irreversible for all operations. Are you sure? [confirm]
 The following will be deleted as a part of factory reset:
 1: Crash info and logs
 2: User data, startup and running configuration
 3: All IOS images, including the current boot image
 4: OBFL logs
 5: User added rommon variables
 6: Data on Field Replaceable Units(USB/SSD/SATA)
 The system will reload to perform factory reset.
 It will take some time to complete and bring it to rommon.
 You will need to load IOS image using USB/TFTP from rommon after
 this operation is completed.
 DO NOT UNPLUG THE POWER OR INTERRUPT THE OPERATION
 Are you sure you want to continue? [confirm]
Chassis 1 reloading, reason - Factory Reset

 Protection key not found
9300L#Oct 25 09:53:05.740: %PMAN-5-EXITACTION: F0/0: pvp: Process manager is exiting:
reload fp action requested
Oct 25 09:53:07.277: %PMAN-5-EXITACTION:vp: Process manager is exiting: rp processes
exit with reload switch code


Enabling factory reset for this reload cycle
 Switch booted with
tftp://10.5.40.45/cat9k_iosxe.BLD_POLARIS_DEV_LATEST_20191007_224933_V17_2_0_21_2.SSA.bin

 Switch booted via
//10.5.40.45/cat9k_iosxe.BLD_POLARIS_DEV_LATEST_20191007_224933_V17_2_0_21_2.SSA.bin
% FACTORYRESET - Started Cleaning Up...

% FACTORYRESET - Unmounting sd1
% FACTORYRESET - Cleaning Up sd1 [0]
% FACTORYRESET - erase In progress.. please wait for completion...
% FACTORYRESET - write zero...
% FACTORYRESET - finish erase

% FACTORYRESET - Making File System sd1 [0]
Discarding device blocks: done
Creating filesystem with 409600 4k blocks and 102544 inodes
Filesystem UUID: fcf01664-7c6f-41ce-99f0-6df1d941701e
Superblock backups stored on blocks:
 32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting information: done

% FACTORYRESET - Mounting Back sd1 [0]
% FACTORYRESET - Handling Mounted sd1
% FACTORYRESET - Factory Reset Done for sd1
```

```
% FACTORYRESET - Unmounting sd3
% FACTORYRESET - Cleaning Up sd3 [0]
% FACTORYRESET - erase In progress.. please wait for completion...
% FACTORYRESET - write zero...


Chassis 2 reloading, reason - Factory Reset
Dec 12 01:02:12.500: %PMAN-5-EXITACTION: F0/0: pvp: Process manager is exiting: reload
fp action requested
De
Enabling factory reset for this reload cycle
 Switch booted with
tftp://10.5.40.45/cat9k_iosxe.BLD_POLARIS_DEV_LATEST_20191007_224933_V17_2_0_21_2.SSA.bin

 Switch booted via
//10.5.40.45/cat9k_iosxe.BLD_POLARIS_DEV_LATEST_20191007_224933_V17_2_0_21_2.SSA.bin
% FACTORYRESET - Started Cleaning Up...
% FACTORYRESET - Unmounting sd1
% FACTORYRESET - Cleaning Up sd1 [0]
% FACTORYRESET - erase In progress.. please wait for completion...
% FACTORYRESET - write zero...


After this the switch will come to boot prompt. Then the customer has to boot the device
 from TFTP.
```

初期設定へのリセットの実行に関する追加情報

関連資料

関連項目	マニュアル タイトル
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。	コマンド リファレンス

初期設定へのリセットに関する機能履歴

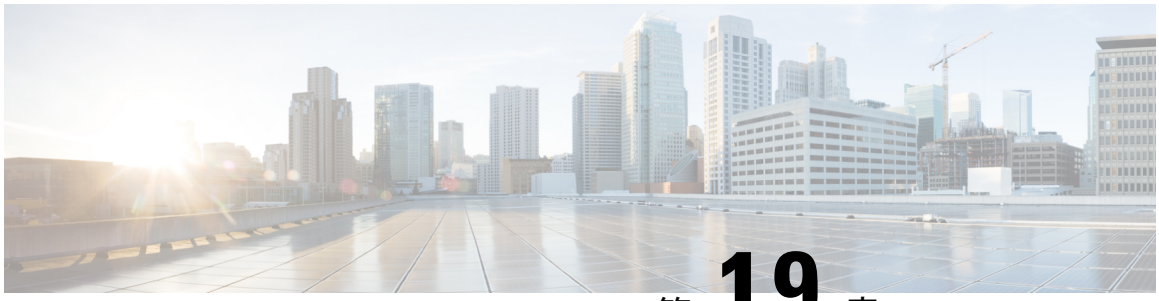
次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

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

リリース	機能	機能情報
Cisco IOS XE Fuji 16.8.1a	工場出荷時の状態へのリセット（Factory Reset）	初期設定にリセットすると、デバイスに保存されているお客様固有のデータがすべて消去され、デバイスの設定は出荷時の元の設定に復元されます

リリース	機能	機能情報
Cisco IOS XE Gibraltar 16.12.1	リムーバブル ストレージ デバイスの初期設定へのリセット	初期設定へのリセットを実行すると、SATA、SSD、USB などのリムーバブル ストレージ デバイスの内容が消去されます。
Cisco IOS XE Amsterdam 17.2.1	3-pass 上書きによる初期設定へのリセット	初期設定へのリセットを実行すると、デバイスからすべてのコンテンツを 3-pass 上書きで安全に消去できます。**secure 3-pass** キーワードが導入されました。
	スタックおよびCisco StackWise Virtual の初期設定へのリセット オプションの拡張	スタック構成デバイスおよびCisco StackWise Virtual 対応デバイスで初期設定へのリセットのサポートが導入されました。

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



第 19 章

セキュア ストレージの設定

セキュア ストレージについて

セキュア ストレージ機能では、重要な設定情報を暗号化して保護できます。非対称キーペア、事前共有秘密、タイプ 6 のパスワード暗号化キーおよび特定のクレデンシャルを暗号化します。インスタンス固有の暗号キーは、危険にさらされることを防ぐためにハードウェアのトラスト アンカーに保管されます。

セキュア ストレージの有効化

始める前に

この機能はデフォルトで無効になっています。

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	**service private-config-encryption** 例： `DEvice(config)# service private-config-encryption`	デバイスでセキュアストレージ機能を有効にします。

	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。
ステップ 4	**write memory** 例： `Device# write memory`	private-config ファイルを暗号化し、暗号化フォーマットで保存します。

セキュア ストレージの無効化

始める前に

デバイスでセキュアストレージ機能を無効にするには、次のタスクを実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**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 service private-config-encryption** 例： `Device(config)# no service private-config-encryption`	デバイスでセキュリティストレージ機能を無効にします。セキュアストレージを無効にすると、すべてのユーザーデータがプレーンテキストで NVRAM に保存されます。
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	**write memory** 例： `Device# write memory`	private-config ファイルを復号し、プレーン フォーマットで保存します。

暗号化のステータスの確認

暗号化のステータスを確認するには、**show parser encrypt file status** コマンドを使用します。次のコマンド出力は、機能は利用できるが、ファイルが暗号化されていないことを示します。ファイルは「プレーン テキスト」形式です。

```
Device#show parser encrypt file status
Feature: Enabled
File Format: Plain Text
Encryption Version: Ver1
```

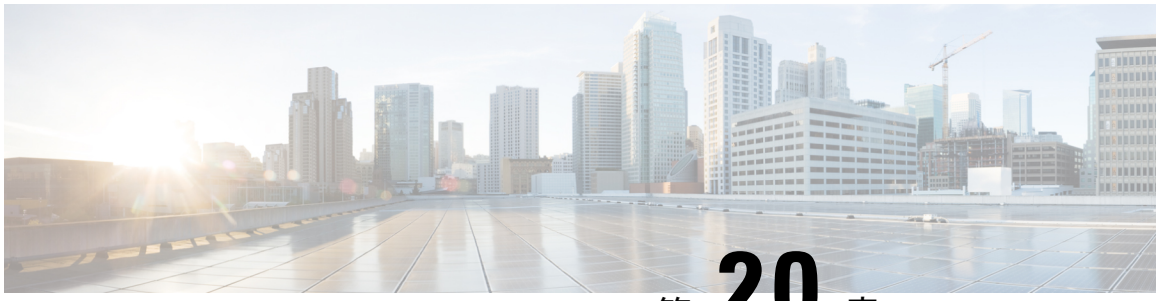
セキュアストレージの機能情報

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

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

リリース	機能	機能情報
Cisco IOS XE Everest 16.5.1a	セキュアなストレージ	セキュア ストレージ機能では、重要な設定情報を暗号化して保護できます。非対称キーペア、事前共有秘密、タイプ 6 のパスワード暗号化キーおよび特定のクレデンシャルを暗号化します。インスタンス固有の暗号キーは、危険にさらされることを防ぐためにハードウェアのトラスト アンカーに保管されます。

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



第 20 章

条件付きデバッグとラジオアクティブ トレース

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。ユーザーは別のレベルに変更することはできません。

ラジオアクティブトレースでは、次の機能が有効になっています。

- IGMP スヌーピング
- レイヤ 2 マルチキャスト

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 サイズの制限に達すると、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 モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**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）。
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）。

	コマンドまたはアクション	目的
ステップ 6	**request platform software trace archive** [**last** {*number*} **days**] [**target** {*crashinfo:* | *flashinfo:*}] 例： `# 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。
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** コマンドと **show platform software trace filter-binary** コマンドは同様の動作をします。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。コマンド **show platform software trace filter-binary** 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#**request platform software trace archive last 1 day**

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 サーバー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ーリング

以下の表に、条件付きデバッグのモニター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。

コマンド	目的
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	マニュアル タイトル
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 9300* シリーズ スイッチ）

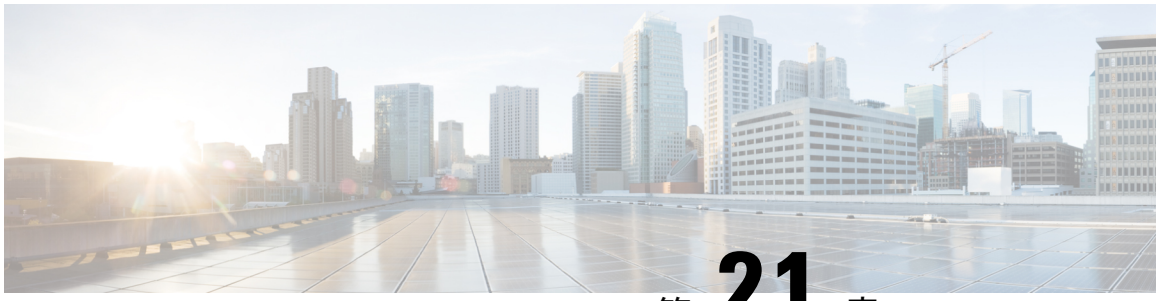
条件付きデバッグとラジオアクティブトレースの機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

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

リリース	機能	機能情報
Cisco IOS XE Everest 16.5.1a	条件付きデバッグとラジオアクティブ トレース	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。

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



第 21 章

同意トークン

同意トークンの制約事項

- 同意トークンはデフォルトで有効であり、無効にすることはできません。
- デバイスからチャレンジが送信された後、30分以内に応答を入力する必要があります。入力しないとチャレンジが期限切れになり、新しいチャレンジの要求が必要になります。
- 単一の応答は、対応するチャレンジに対して1回だけ有効です。
- ルートシェルアクセスの最大承認タイムアウトは7日間です。
- スイッチオーバーイベント後、既存の同意トークンベースの承認はすべて期限切れとして処理されます。その後、サービスアクセスの新しい認証シーケンスを再起動する必要があります。
- シスコのチャレンジ署名サーバー上の同意トークン応答生成にアクセスできるのは、シスコ認定担当者のみです。
- システムシェルアクセスのシナリオでは、承認タイムアウトが発生するか、または同意トークン終了承認コマンドによってシェル承認が明示的に終了されるまで、シェルを終了しても承認は終了しません。

  システムシェルアクセスの目的を達成したら、同意トークン終了コマンドを明示的に発行することによって、システムシェルの承認を強制終了することを推奨します。

同意トークンに関する情報

同意トークンは、ネットワーク管理者と Cisco Technical Assistance Centre（Cisco TAC）の相互の同意により、システムシェルにアクセスする組織のネットワーク管理者を認証するために使用されるセキュリティ機能です。

一部のデバッグシナリオでは、Cisco TAC エンジニアが特定のデバッグ情報を収集したり、実稼働システムでライブデバッグを実行する必要がある場合があります。このような場合、Cisco TACエンジニアは、デバイスのシステムシェルにアクセスするようユーザー（ネットワーク管理者）に依頼します。同意トークンは、システムシェルへの特権アクセス、制限アクセス、およびセキュアアクセスを提供する、ロック、ロック解除、および再ロックのメカニズムです。

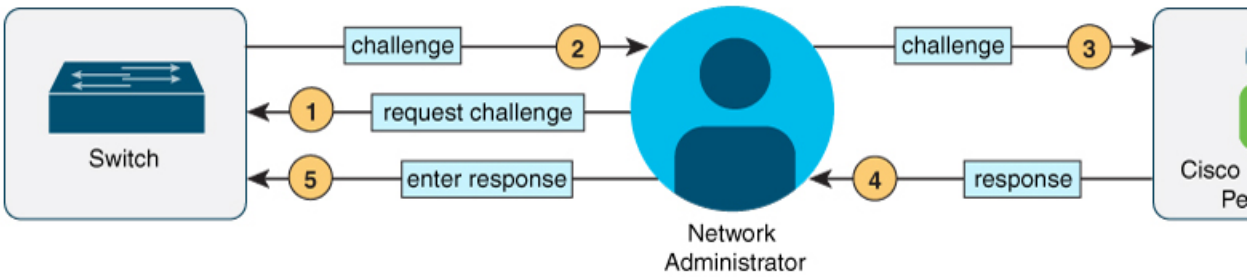
システムシェルへのアクセスを要求する場合は、認証を受ける必要があります。最初にコマンドを実行し、デバイスの同意トークン機能を使用してチャレンジを生成する必要があります。デバイスは、固有のチャレンジを出力として生成します。このチャレンジ文字列をコピーし、電子メールまたはインスタントメッセージでシスコ認定担当者に送信する必要があります。

シスコ認定担当者は、一意のチャレンジ文字列を処理し、一意のレスポンスを生成します。シスコ認定担当者はこのレスポンス文字列をコピーし、電子メールまたはインスタントメッセージで送信します。

次に、このレスポンス文字列をデバイスに入力する必要があります。チャレンジ/レスポンスペアが一致すると、システムシェルへのアクセスが許可されます。一致しない場合は、エラーが表示され、認証プロセスを繰り返す必要があります。

システムシェルにアクセスしたら、Cisco TAC エンジニアが必要とするデバッグ情報を収集します。システムシェルへのアクセスが完了したら、セッションを終了し、デバッグプロセスを続行します。

図 *15* : 同意トークン



システムシェルアクセスの同意トークン承認プロセス

ここでは、システムシェルにアクセスするための同意トークン承認のプロセスについて説明します。

手順

ステップ 1 指定された期間、システムシェルへのアクセスを要求するチャレンジを生成します。

例：

```
Device# request consent-token generate-challenge shell-access auth-timeout 900
[redacted]
Device#
*Jan 18 02:47:06.733: %CTOKEN-6-AUTH_UPDATE: Consent Token Update (challenge generation
 attempt: Shell access 0).
```

request consent-token generate-challenge shell-access *time-validity-slot* コマンドを使用して、チャレンジの要求を送信します。システムシェルへのアクセスを要求する期間（分単位）は、time-slot-period です。

この例の期間は、セッションの期限切れ後 900 分です。

デバイスは、固有のチャレンジを出力として生成します。このチャレンジは、base-64 形式の文字列です。

ステップ 2 シスコ認定担当者にチャレンジ文字列を送信します。

デバイスによって生成されたチャレンジ文字列を、電子メールまたはインスタントメッセージでシスコ認定担当者に送信します。

シスコ認定担当者は固有のチャレンジ文字列を処理し、レスポンスを生成します。レスポンスもまた、固有の base-64 文字列です。シスコ認定担当者はこのレスポンス文字列をコピーし、電子メールまたはインスタントメッセージで送信します。

ステップ 3 デバイスにレスポンス文字列を入力します。

例：

```
Device# request consent-token accept-response shell-access
[redacted]
% Consent token authorization success
*Jan 18 02:51:37.807: %CTOKEN-6-AUTH_UPDATE: Consent Token Update (authentication success:
 Shell access 0).

Device# request platform software system shell
Activity within this shell can jeopardize the functioning of the system.
Are you sure you want to continue? [y/n] y
Device#
*Jan 18 02:56:59.714: %CTOKEN-6-AUTH_UPDATE: Consent Token Update (authorization for
Shell access 0 will expire in 10 min).
```

request consent-token accept-response shell-access *response-string* コマンドを使用して、シスコ認定担当者から送信されたレスポンス文字列を入力します。

チャレンジ/レスポンスペアが一致すると、システムシェルへのアクセスが許可されます。チャレンジ/レスポンスペアが一致しない場合は、エラーが表示され、手順 1 ー 3 を繰り返す必要があります。

承認されると、要求されたタイムスロットのシステムシェルにアクセスできます。

承認セッションの残り時間が 10 分になると、デバイスはメッセージを送信します。

ステップ 4 セッションを終了します。

例：

```
Device# request consent-token terminate-auth
% Consent token authorization termination success

Device#
*Jan 18 23:33:02.937: %CTOKEN-6-AUTH_UPDATE: Consent Token Update (terminate
authentication: Shell access 0).
Device#
```

システムシェルへのアクセスが終了したら、 **request consent-token terminate-auth** コマンドを使用してセッションを終了できます。このコマンドを使用して、承認タイムアウトの前にセッションを強制終了することもできます。要求したタイムスロットが期限切れになると、セッションも自動的に終了します。

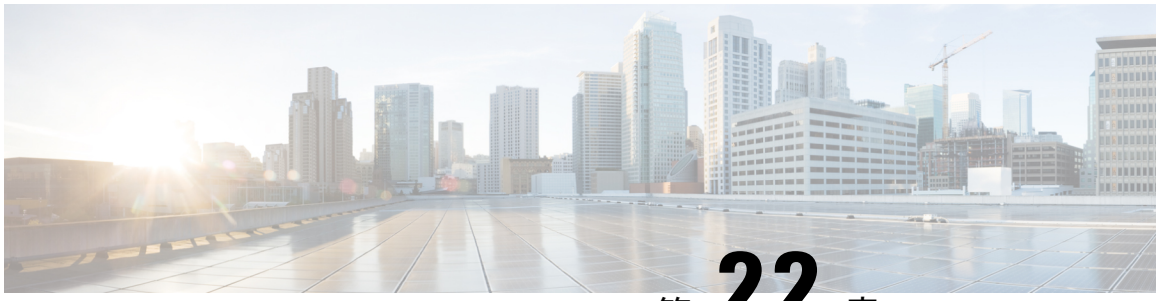
同意トークンの機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

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

リリース	機能	機能情報
Cisco IOS XE Gibraltar 16.11.1	同意トークン	同意トークンは、ネットワーク管理者と Cisco Technical Assistance Centre（Cisco TAC）の相互の同意により、システムシェルにアクセスする組織のネットワーク管理者を認証するために使用されるセキュリティ機能です。

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



第 22 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（536 ページ）の項で説明されている手順に従います。

デバイスのパスワードを紛失したか忘れた場合

デバイスのデフォルト設定では、デバイスを直接操作するエンドユーザーが、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、デバイスを直接操作してください。

（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場合に限りエンドユーザー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、エンドユーザー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 （540 ページ）の項で説明する手順に従います。

Power over Ethernet（PoE）ポート

Power over Ethernet（PoE）スイッチ ポートでは、回路に電力が供給されていないことをスイッチが検知した場合、接続している次のデバイスに電力が自動的に供給されます。

- シスコ先行標準受電デバイス（Cisco IP Phone や Cisco Aironet アクセス ポイントなど）
- IEEE 802.3af 準拠の受電装置
- IEEE 802.3at 準拠の受電装置

受電デバイスが PoE スイッチポートおよび AC 電源に接続されている場合、冗長電力として利用できます。受電デバイスが PoE ポートにだけ接続されている場合、受電デバイスには冗長電力は供給されません。

受電デバイスを検出すると、スイッチは受電デバイスの電力要件を判断し、受電デバイスへの電力供給を許可または拒否します。また、スイッチは消費電力をモニターリングおよびポリシングすることで、装置の電力の消費をリアルタイムに検知できます。

詳細については、『*Interface and Hardware Component Configuration Guide (Catalyst 9300 Switches)*』の「Configuring PoE」の章を参照してください。

PoE のさまざまなトラブルシューティング シナリオについては、Power over Ethernet（PoE）に関するトラブルシューティングのシナリオ （552 ページ）の項を参照してください。

電力消失によるポートの障害

PoE デバイスポートに接続され、AC 電源から電力が供給されている受電デバイス（Cisco IP Phone 7910 など）に AC 電源から電力が供給されない場合、そのデバイスは errdisable ステートになることがあります。errdisable ステートから回復するには、**shutdown** 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、**no shutdown** インターフェイスコマンドを入力します。デバイスで自動回復を設定し、errdisable ステートから回復することもできます。

デバイスの場合、**errdisable recovery cause loopback** および **errdisable recovery interval** *seconds* グローバル コンフィギュレーション コマンドは、指定した期間が経過したあと自動的にインターフェイスを errdisable ステートから復帰させます。

不正リンク アップによるポート障害

シスコ受電デバイスをポートに接続し、**power inline never** 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設定した場合は、不正リンクアップが発生し、ポートが errdisable ステートになることがあります。ポートを errdisable ステートから回復するには、**shutdown** および **no shutdown** インターフェイス コンフィギュレーション コマンドを入力します。

power inline never コマンドで設定したポートにシスコ受電デバイスを接続しないでください。

ping

デバイスは IP の ping をサポートしており、これを使用し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 （546 ページ）の項を参照してください。

レイヤ 2 トレースルート

レイヤ 2 トレースルート機能により、パケットが通過する送信元デバイスから宛先デバイスまでの物理パスを識別できます。レイヤ 2 トレースルートは、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。トレースルートは、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トレースルート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トレースクエリを送信し続け、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 を無効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であると定義できます。物理パス内のすべてのデバイスは、他のデバイス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。

- 指定された送信元および宛先アドレスが同じ VLAN にある場合、**traceroute mac** コマンド出力はレイヤ 2 パスを表示します。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。

- 指定された送信元および宛先の IP アドレスが同一のサブネット内にある場合、**traceroute mac ip** コマンド出力はレイヤ 2 パスを表示します。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 を対応させます。

  - 指定の IP アドレスの ARP のエントリが存在している場合、デバイスは関連付けられた MAC アドレスを使用し、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 アドレスを解決しようと試みます。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。
- レイヤ2トレースルートは、ユーザー データグラム プロトコル（UDP）ポート2228でリスニングソケットを開きます。このポートは、任意のIPv4アドレスを使用してリモートからアクセスでき、認証は必要ありません。このUDPソケットにより、VLAN情報、リンク、特定のMACアドレスの存在、およびCDPネイバー情報をデバイスから読み取ることができます。この情報を使用することにより、最終的にレイヤ2ネットワークトポロジの全体像を構築できます。
- レイヤ2トレースルートはデフォルトで有効になっており、グローバル コンフィギュレーション モードで **no l2 traceroute** コマンドを実行することによって無効にできます。レイヤ 2 トレースルートを再度有効にするには、グローバル コンフィギュレーション モードで **l2 traceroute** コマンドを使用します。

IP トレースルート

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、**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、このデバイスは 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ー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、ユーザー 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 （558 ページ）に進み、IP traceroute プロセスの例を参照してください。

Time Domain Reflector ガイドライン

Time Domain Reflector（TDR）機能を使用すると、ケーブル配線の問題を診断して解決できます。TDR 稼働時、ローカル デバイスはケーブルを介して信号を送信して、最初に送信した信号と反射された信号を比べます。

TDR は次のケーブル障害を検出します。

- ツイストペア ケーブルの導線のオープン、損傷、切断：導線がリモート デバイスからの導線に接続されていない状態。
- ツイストペア ケーブルの導線のショート：導線が互いに接触している状態、またはリモート デバイスからの導線に接触している状態。たとえば、ツイスト ペア ケーブルの一方の導線が、もう一方の導線にはんだ付けされている場合、ツイストペア ケーブルのショートが発生します。

  ツイストペアの導線の一方がオープンになっている場合、TDR はオープンになっている導線の長さを検出できます。

次の状況で TDR を使用して、ケーブル障害を診断および解決してください。

- デバイスの交換
- 配線クローゼットの設定
- リンクが確立できない、または適切に動作していない場合における、2 つのデバイス間の接続のトラブルシューティング

TDR の実行時、次の場合にデバイスは正確な情報をレポートします。

- ギガビット リンク用のケーブルが単線コア ケーブル
- オープンエンド ケーブルが未終端

TDR の実行時、次の場合にデバイスは正確な情報をレポートしません。

- ギガビット リンク用のケーブルがツイストペア ケーブルまたは連続接続された単線コア ケーブル
- リンクが 10 Mb または 100 Mb

- より線ケーブル
- リンク パートナーが Cisco IP Phone
- リンク パートナーが IEEE 802.3 に準拠していない

TDR の実行および結果の表示 （548 ページ）に移動し、TDR のコマンドを確認します。

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ー数が少ない期間に **debug** コマンドを使用することをお勧めします。デバッギン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低くなります。

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メンバーで生成されます。スタック内の他のメンバーではレポートは生成されません。
- 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。

コアダンプを生成するには、**request platform software process core fed active** コマンドを使用します。

```
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 サーバー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]?
```

スタックの全メンバーからのトレースログは、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。

複雑なネットワークでは、システムレポートファイルの送信元を追跡することは困難です。システムレポートファイルが一意に識別できる場合、この作業は簡単になります。Cisco IOS XE Amsterdam 17.3.x リリース以降、システムレポートファイル名の前にホスト名が追加され、レポートが一意に識別できるようになります。

次の例では、ホスト名が先頭に追加されたシステムレポートファイルを表示します。

```
HOSTNAME#dir flash:/core | grep HOSTNAME
40486  -rw-        108268293  Oct 21 2019 16:07:50 -04:00
HOSTNAME-system-report_20191021-200748-UTC.tar.gz
40487  -rw-            17523  Oct 21 2019 16:07:56 -04:00
```

```
HOSTNAME-system-report_20191021-200748-UTC-info.txt
40484  -rw-         48360998  Oct 21 2019 16:55:24 -04:00
HOSTNAME-system-report_20191021-205523-UTC.tar.gz
40488  -rw-            14073  Oct 21 2019 16:55:26 -04:00
HOSTNAME-system-report_20191021-205523-UTC-info.txt
```

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またはスイッチスタックメンバに入力された 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 またはスイッチ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またはスイッチスタックメンバ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。
- Power over Ethernet（PoE）：スタンドアロンデバイスまたはスイッチスタックメンバの 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またはスイッチスタックメンバの温度。
- 稼働時間：スタンドアロンデバイスまたはスイッチスタックメンバが起動された際の時刻、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 またはスイッチスタックメンバのシステム電圧。

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 つ目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。

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。

スイッチのコンソールポートのデフォルトレートである 9600 ビット/秒（bps）と一致するように、端末のボーレートを設定します。ボーレートが 9600 bps 以外の値に設定されている場合、速度がデフォルトに戻るまでコンソールへのアクセスは失われます。

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 サーバーにソフトウェア 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。

ステップ 6 ブートローダープロンプトで、TFTP サーバーに ping を実行できることを確認します。

a) スイッチの IP アドレスを設定します：**set IP_ADDRESS** *ip_address*

例：

```
switch: set IP_ADDRESS 192.0.2.123
```

b) スイッチのサブネットマスクを設定します：**set IP_SUBNET_MASK** *subnet_mask*

例：

```
switch: set IP_SUBNET_MASK 255.255.255.0
```

c) デフォルトゲートウェイを設定します： **set DEFAULT_GATEWAY** *ip_address*

例：

```
switch: set DEFAULT_ROUTER 192.0.2.1
```

d) 次のコマンドを実行して、TFTP サーバーに 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 次のいずれかを選択します。

- ブートローダープロンプトで、**boot tftp** コマンドを開始します。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。

```
switch: boot tftp://10.168.0.1/cat9k/cat9k_iosxe.2017-08-25_09.41.bin
attempting to boot from [tftp://10.168.0.1/cat9k/cat9k_iosxe.2017-08-25_09.41.SSA.bin]


 interface : eth0
 macaddr   : E4:AA:5D:59:7B:44
 ip        : 10.168.247.10
 netmask   : 10.255.0.0
 gateway   : 10.168.0.1
 server    : 10.168.0.1
 file      : cat9k/cat9k_iosxe.2017-08-25_09.41.bin
```

```
             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
16.6.1 RELEASE SOFTWARE (fc2)
Copyright (c) 1986-2017 by Cisco Systems, Inc.
Compiled Thu 24-Aug-17 13:23 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 C9XXX (X86) processor (revision V00) with 869398K/6147K bytes of memory.
Processor board ID FXS1939Q3LZ
144 Gigabit Ethernet interfaces
16 Ten Gigabit Ethernet interfaces
4 Forty Gigabit Ethernet interfaces
32768K bytes of non-volatile configuration memory.
15958516K bytes of physical memory.
11161600K bytes of Bootflash at bootflash:.
1638400K bytes of Crash Files at crashinfo:.
0K bytes of WebUI ODM Files at webui:.

%INIT: waited 0 seconds for NVRAM to be available



Press RETURN to get started!
```

- リカバリパーティションからソフトウェアをインストールします。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。

a) 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。

例：

```
switch: dir sda9:

Size            Attributes      Name
 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
21680202        -rw-            cat9k-recovery.SSA.bin
 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

b) ブートローダープロンプトで、emergency-install 機能を開始します。この機能を使用すると、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。

例：

```
switch: emergency-install
tftp://10.255.254.254/auto/tftpboot/X86/cat9k_iosxe.16.05.01a.SPA.bin
WARNING: The system partition (bootflash:) will be erased during the system recovery
 install process.
Are you sure you want to proceed? [y]  y/n [n]: y
Starting system recovery
(tftp://10.255.254.254/auto/tftpboot/X86/cat9k_iosxe.16.05.01a.SPA.bin) ...
```

```
Attempting to boot from [sda9:cat9k-recovery.SSA.bin]
Located cat9k-recovery.SSA.bin
############################################################################################################################

Warning: ignoring ROMMON var "BOOT_PARAM"

PLATFORM_TYPE C9X00 speed 9600

Booting Recovery Image 16.5.1a



Initiating Emergency Installation of bundle
tftp://10.255.254.254/auto/tftpboot/X86/cat9k_iosxe.16.05.01a.SPA.bin


Downloading bundle
tftp://10.255.254.254/auto/tftpboot/X86/cat9k_iosxe.16.05.01a.SPA.bin...
curl_vrf=2
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  485M  100  485M    0     0  5143k      0  0:01:36  0:01:36 --:--:-- 5256k
100  485M  100  485M    0     0  5143k      0  0:01:36  0:01:36 --:--:-- 5143k

Validating bundle
tftp://10.255.254.254/auto/tftpboot/X86/cat9k_iosxe.16.05.01a.SPA.bin...
Installing bundle
tftp://10.255.254.254/auto/tftpboot/X86/cat9k_iosxe.16.05.01a.SPA.bin....
Verifying bundle
tftp://10.255.254.254/auto/tftpboot/X86/cat9k_iosxe.16.05.01a.SPA.bin...
Package cat9k-cc_srdriver.16.05.01a.SPA.pkg
/temp//stage/cat9k-cc_srdriver.16.05.01a.SPA.pkg is Digitally Signed
Package cat9k-espbase.16.05.01a.SPA.pkg /temp//stage/cat9k-espbase.16.05.01a.SPA.pkg
 is Digitally Signed
Package cat9k-guestshell.16.05.01a.SPA.pkg
/temp//stage/cat9k-guestshell.16.05.01a.SPA.pkg is Digitally Signed
Package cat9k-rpbase.16.05.01a.SPA.pkg /temp//stage/cat9k-rpbase.16.05.01a.SPA.pkg
is Digitally Signed
Package cat9k-sipbase.16.05.01a.SPA.pkg /temp//stage/cat9k-sipbase.16.05.01a.SPA.pkg
 is Digitally Signed
Package cat9k-sipspa.16.05.01a.SPA.pkg /temp//stage/cat9k-sipspa.16.05.01a.SPA.pkg
is Digitally Signed
Package cat9k-srdriver.16.05.01a.SPA.pkg /temp//stage/cat9k-srdriver.16.05.01a.SPA.pkg
 is Digitally Signed
Package cat9k-webui.16.05.01a.SPA.pkg /temp//stage/cat9k-webui.16.05.01a.SPA.pkg is
 Digitally Signed
Package cat9k-wlc.16.05.01a.SPA.pkg /temp//stage/cat9k-wlc.16.05.01a.SPA.pkg is
Digitally Signed
Package /cat9k-rpboot.16.05.01a.SPA.pkg /temp//rpboot/cat9k-rpboot.16.05.01a.SPA.pkg
 is Digitally Signed
Preparing flash....
Flash filesystem unmounted successfully /dev/sdb3
Syncing device....
Emergency Install successful... Rebooting
Will reboot now

Initializing Hardware...

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

Current image running:
Primary Rommon Image
```

```
Last reset cause: SoftwareReload
C9X00 platform with 8388608 Kbytes of main memory
```

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ーが、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ー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 ユーザー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。スイッチスタックのパスワードを回復する場合は、アクティブ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。スイッチ スタックのパスワードを回復する場合は、スタック メンバの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 スイッチまたはアクティブスイッチに電源コードを再接続します。システム LED が点滅したら、すぐに [Mode] ボタンを 2 ー 3 回押して放します。スイッチは ROMMON モードを開始します。

リロード中に次のコンソールメッセージが表示されます。

```
Initializing Hardware...

System Bootstrap, Version 16.6.1r [FC1], RELEASE SOFTWARE (P)
Compiled Sat 07/15/2017  8:31:57.39 by rel

Current image running:
Primary Rommon Image
```

```
Last reset cause: SoftwareReload         <---- Start pressing and releasing the mode
button
C9300-24U platform with 8388608 Kbytes of main memory

attempting to boot from [flash:packages.conf]

Located file packages.conf
#
############################################################################

Unable to load cat9k-rpboot.16.06.02b.SPA.pkg
Failed to boot file flash:user/packages.conf
ERROR: failed to boot from flash:packages.conf (Aborted) <--- will abort
switch:
switch:  <---- ROMMON
```

「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
```

アクティブ スイッチの場合

```
Switch> reload slot <stack-active-member-number>
Proceed with reload? [confirm] y
```

ステップ 6 スタック内の残りのスイッチに電源を投入します。

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。

```
Device: 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。

```
Device: 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。

```
Device> enable
Device#
```

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。

```
Device# 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。

```
Device# configure terminal
Device(config)# enable secret password
```

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
```

ステップ 9 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。

```
Device# show boot

 BOOT variable = flash:packages.conf;
 Manual Boot = yes
 Enable Break = yes
```

ステップ 10 デバイスのリロード。

```
Device# reload
```

ステップ 11 SWITCH_IGNORE_STARTUP_CFG パラメータを 0 に設定します。

```
Device(config)# no system ignore startupconfig switch all
Device(config)# end
Device# write memory
```

ステップ 12 フラッシュの *packages.conf* ファイルを使用して、デバイスを起動します。

```
Device: boot flash:packages.conf
```

ステップ 13 デバイスが起動したら、デバイスで手動ブートを無効にします。

```
Device(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と 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。スイッチは、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のステータスを確認し、error-disabled 状態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 状態からインターフェイスを回復させ、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。

SFP モジュール ステータスのモニターリング

show interfaces transceiver 特権 EXEC コマンドを使用すると、SFP モジュールの物理または動作ステータスを確認できます。このコマンドは、温度や特定のインターフェイス上の SFP モジュールの現状などの動作ステータスと、アラーム ステータスを表示します。また、このコマンドを使用してSFPモジュールの速度およびデュプレックス設定も確認できます。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にある **show interfaces transceiver** コマンドを参照してください。

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。

温度のモニターリング

デバイスは温度条件をモニターし、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。

物理パスのモニターリング

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ーできます。

表 *28 :* 物理パスのモニターリング

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 *host* `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は、インターフェイス上で実行する場合、アクティブスイッチ上でもスタックメンバ上でも実行できます。

TDR を実行するには、**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

デフォルトでは、ネットワークサーバーが **debug** コマンドからの出力とシステムエラーメッセージ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ーできます。

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サーバーを実行しているUNIXホストがあります。Syslogフォーマットは、4.3BSD 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サーバー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は、インターフェイスに着信するパケットがシステムを介して送信された場合の転送結果に関する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、デバイスの特定用途向け集積回路（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、ユーザーも少ないときに使用することを推奨します。デバッギン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低くなります。

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。

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 *29 : 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。

コマンド	目的
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。
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 *30 : 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	修正措置
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 *31 : 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** ユーザー 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。 該当するインターフェイスまたはポートに **power inline never** が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） シスコ受電装置は、ストレート ケーブルでのみ機能します。クロスオーバ ケーブルでは機能しません。 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 power inline** コマンドを使用して、利用可能な電力量を確認します。

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がないか、**show log** 特権EXECコマンドを使用して調べます。 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、またはerror-disabled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、ポートを再度有効にします。 **show env power** および **show power inline** 特権EXECコマンドを使用して、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されていない場合に、受電デバイスに電

症状または問題	考えられる原因と解決法
	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際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、インラインパワーの統計情報とポートのステータスをモニターします。 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装置は、切断またはリセットされます。 正常に動作した後で、シスコ電話機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）が枯渇していないか確認します。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がスイッチに検出され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 *32 : 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 サーバーのタイムアウトが 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	ユーザー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 *33 : 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に関する追加情報

関連資料

関連項目	マニュアル タイトル
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 9300* シリーズ スイッチ）

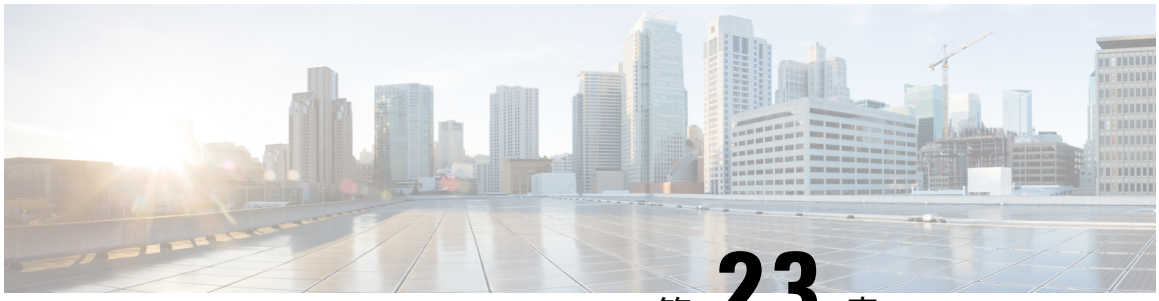
ソフトウェア設定のトラブルシューティングの機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

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

リリース	機能	機能情報
Cisco IOS XE Everest 16.5.1a	ソフトウェア設定のトラブルシューティング	ソフトウェア設定のトラブルシューティングでは、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。
Cisco IOS XE Amsterdam 17.3.1	システムレポートファイル	ホスト名がシステムレポートファイルの先頭に追加されます。これにより、システムレポートファイルが一意に識別可能になります。

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



第 23 章

回線の自動統合

回線の自動統合

Cisco IOS XE ソフトウェアは、不揮発性生成（NVGEN）プロセスを実行して、デバイスの設定状態を取得します。NVGEN プロセス中に、システムは共通のパラメータに基づいて line コマンドを自動的に統合します。

デバイスが Cisco Digital Network Architecture（DNA）センターまたは Cisco vManage に接続し、Yet Another Next Generation（YANG）インターフェイスを介して回線設定を送信すると、設定が自動統合されます。これにより、デバイスと DNA Center の間に不一致が生じる可能性があります。設定の不一致により、デバイスから DNA Center への逆同期が発生する場合もあります。この逆同期の間、デバイスは他の設定変更の影響を受けないようにロックされます。その結果、デバイスのパフォーマンスに影響が及ぶ可能性があります。

Cisco IOS XE 17.4.1 リリース以降では、グローバル コンフィギュレーション モードで **no line auto-consolidation** コマンドを使用して、line コマンドの自動統合を無効にできます。自動統合は、デフォルトでは有効になっています。無効にするには、このコマンドの no 形式を使用します。

デバイスでの設定を表示するには、**show running-configuration all** コマンドを使用します。次の例では、line auto-consolidation が有効になっています。

```
Device#sh running-config all | i auto-consolidation
line auto-consolidation
```

自動統合を無効にすると、**show run** コマンドの出力が非常に長くなります。この点は、実行コンフィギュレーション ファイルとスタートアップ コンフィギュレーション ファイルのサイズに影響します。自動統合を無効にすると、次の動作が発生します。

- サブモードで同じ設定に属する回線の連続的なグループが単一の範囲内にまとめられることがなくなります。

  ```
  Device#show run | sec line
  line con 0
  stopbits 1
  ```

```
line vty 0 4
transport input ssh
line vty 5 9
transport input all
Device#configure terminal
Device(config)#no line auto-consolidation
Device(config)#line vty 10 15
Device(config-line)#transport input all
Device(config-line)#end
Device#show run | sec line
no line auto-consolidation
line con 0
stopbits 1
line vty 0 4
transport input ssh
line vty 5 9
transport input all
line vty 10 15
transport input all
```

- 自動統合を有効にして一部の回線を設定した後に自動統合を無効にすると、自動統合を無効にした後に設定された回線のみが統合されません。

```
Device#show run | sec line
line con 0
stopbits 1
line vty 0 4
transport input ssh
line vty 5 9
transport input all
Device#configure terminal
Device(config)#line vty 10 15
Device(config-line)#transport input all
Device(config-line)#end
Device#show run | sec line
line con 0
stopbits 1
consolidated line vty 0 4
transport input ssh
line vty 5 15
transport input all
Device#configure terminal
Device(config)#no line auto-consolidation
Device(config)#line vty 16 20
Device(config-line)#transport input all
Device(config-line)#end
Device#show run | sec line
no line auto-consolidation
line con 0
stopbits 1
consolidated line vty 0 4
transport input ssh
line vty 5 15
transport input all
line vty 16 20
transport input all
```

- 自動統合を無効にした後で有効にすると、統合されなかった回線が自動統合されます。

```
Device#sh running-config | sec line
no line auto-consolidation
line con 0
exec-timeout 0 0
logging synchronous
```

```
stopbits 1
line vty 0 4
transport input ssh
line vty 5 15
transport input ssh
line vty 16 19
transport input ssh
Device#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)#line vty 20 25
Device(config-line)#transport input ssh
Device(config-line)#end
Device#sh running-config | sec line
no line auto-consolidation
line con 0
exec-timeout 0 0
logging synchronous
stopbits 1
line vty 0 4
transport input ssh
line vty 5 15
transport input ssh
line vty 16 19
transport input ssh
line vty 20 25
transport input ssh
Device#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)#line auto-consolidation
Device(config)#end
Device#show running-config | sec line
line con 0
exec-timeout 0 0
logging synchronous
stopbits 1
line vty 0 4
transport input ssh
line vty 5 25
transport input ssh
```

- 範囲の連続している回線を設定できます。設定が許可されます。

```
Device#show run | sec line
no line auto-consolidation
line con 0
stopbits 1
line vty 0 4
transport input ssh
Device#configure terminal
Device(config)#line vty 5  20
Device(config)#transport input all
Device(config-line)#end
Device#show run | sec line
line con 0
stopbits 1
line vty 0 4
transport input ssh
line vty 5 20
transport input all
```

- 範囲が連続していない回線は設定できません。設定が拒否されます。

```
Device#show run | sec line
no line auto-consolidation
line con 0
```

回線の自動統合

```
logging synchronous
line aux 0
line vty 0 4
transport input none
Device# configure terminal
Device(config)# line vty 10 20
% Bad line number - VTY line number is not contiguous.
```

- リストの最後にある連続した回線を削除できます。コントローラモードでは、一度に1つの回線を削除できます。回線を一括で削除することはできません。自律モードでは、回線を一括で削除できます。

```
Device# show run | sec line
no line auto-consolidation
line con 0
stopbits 1
line vty 0 4
transport input ssh
line vty 5 20
transport input all
Device# configure terminal
Device(config)# no line vty 5 20
Device(config)# end
Device#show run | sec line
line con 0
stopbits 1
line vty 0 4
transport input ssh
```

- リストの最後にある連続していない回線は削除できません。削除されると連続していない範囲が生じるような回線は削除できません。この操作により、回線を削除できないことを示すエラーメッセージが生成されます。

```
Device# show run | sec line
no line auto-consolidation
line con 0
stopbits 1
line vty 0 4
transport input ssh
line vty 5 9
transport input none
line vty 10 20
transport input all
Device# configure terminal
Device(config)# no line vty 5 9
% Cannot delete the 9 line number as it is not the last VTY line number
```

- 使用中の回線やデフォルトの回線は削除できません。

```
Device#show run | sec line
no line auto-consolidation
line con 0
stopbits 1
line vty 0 4
transport input ssh
line vty 5 20
transport input ssh
Device#configure terminal
Router(config)#no line vty 15
% Can't delete last 16 VTY lines, lines in use, statbit: 0x10C40, tiptop: 590
% process name: SSH Process
```

- 自律モードでは、サブ範囲を変更できます。変更すると回線が分割され、設定の逆同期が発生します。コントローラモードでは、サブ範囲を変更できません。これはコントローラモードと自律モード間の動作の相違点です。コントローラモードでは、コントローラからプッシュされた設定との不一致を回避するために、サブ範囲の変更は拒否されます。

  次の例は、自律モードでサブ範囲を変更する方法を示しています。

```
Device#show run | sec line
no line auto-consolidation
line con 0
stopbits 1
line vty 0 4
transport input ssh
line vty 5 9
transport input none
Device#configure terminal
Device(config)#line vty 7 8
Device(config-line)#transport input telnet
Device(config-line)#end
Device#show run | sec line
line con 0
 stopbits 1
line vty 0 4
 transport input ssh
line vty 5 6
 transport input none
line vty 7 8
 transport input telnet
line vty 9
 transport input none
```

- 次の例は、サブ範囲の変更がコントローラモードでサポートされていないことを示しています。

```
Device#show run | sec line
no line auto-consolidation
line con 0
stopbits 1
line vty 0 4
transport input ssh
line vty 5 9
transport input none
Device#configure terminal
Device(config)# line vty 5 8
Device(config-line)# end
Uncommitted changes found, commit them? [yes/no/CANCEL] yes
Aborted: inconsistent value: Device refused one or more commands:
line vty 5 8
      ^
% Invalid input detected at '^' marker.
Component Response: "
% Modifications of overlapping/sub range is not allowed in controller mode"
Error executing command: CLI command error -
Device(config)# end
```

- 自律モードでは、重複する範囲を変更できます。変更すると回線が分割され、設定の逆同期が発生します。コントローラモードでは、重複する範囲を変更できません。コントローラモードでは、コントローラからプッシュされた設定との不一致を回避するために、重複する範囲の変更は拒否されます。

  次の例は、自律モードで重複する範囲を変更する方法を示しています。

回線の自動統合

```
Device#show run | sec line
no line auto-consolidation
line con 0
stopbits 1
line vty 0 4
transport input ssh
line vty 5 10
transport input none
line vty 11 20
transport input all
Device#configure terminal
Device(config)#line vty 8 12
Device(config-line)#transport input ssh
Device(config-line)#end
Device#show run | sec line
line con 0
stopbits 1
line vty 0 4
transport input ssh
line vty 5 7
transport input none
line vty 8 10
transport input ssh
line vty 11 12
transport input ssh
line vty 13 20
transport input all
```

- 次の例は、重複する範囲の変更がコントローラモードでサポートされていないことを示しています。

```
Device#show run | sec line
no line auto-consolidation
line con 0
stopbits 1
line vty 0 4
transport input ssh
line vty 5 10
transport input none
line vty 11 20
transport input all
Device(config)# line vty 5 11
Device(config-line)# end
Uncommitted changes found, commit them? [yes/no/CANCEL] yes
Aborted: inconsistent value: Device refused one or more commands:
line vty 5 11
      ^
% Invalid input detected at '^' marker.
Component Response: "
% Modifications of overlapping/sub range is not allowed in controller mode"
Error executing command: CLI command error -
Device(config)# end
```

- 自動統合が有効な状態から自動統合が無効な状態に設定を置き換えることができます。

```
Device#show run | sec line
line con 0
stopbits 1
line vty 0 4
transport input all
line vty 5 9
transport input ssh
line vty 10 15
transport input telnet
```

```
line vty 16 20
transport input ssh

Device#configure replace bootflash:cfg2.txt
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#show run | sec line
no line auto-consolidation
line con 0
stopbits 1
line vty 0 4
transport input all
line vty 5 20
transport input ssh
```

- 自動統合が無効な状態から自動統合が有効な状態に設定を置き換えることができます。

```
Device#show run | sec line
no line auto-consolidation
line vty 0 4
transport input all
line vty 5 20
transport input ssh

Device#configure replace bootflash:cfg1.txt
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#show run | sec line
line con 0
stopbits 1
line vty 0 4
transport input all
line vty 5 9
transport input ssh
line vty 10 15
transport input telnet
line vty 16 20
transport input ssh
```

回線の自動統合の機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

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

リリース	機能	機能情報
Cisco IOS XE Bengaluru 17.4.1	回線の自動統合	line コマンドの自動統合は、デフォルトで有効になっています。**no line auto-consolidation** コマンドは、line コマンドの自動統合を無効にするために使用できます。 **line auto-consolidation** コマンドが導入されました。

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