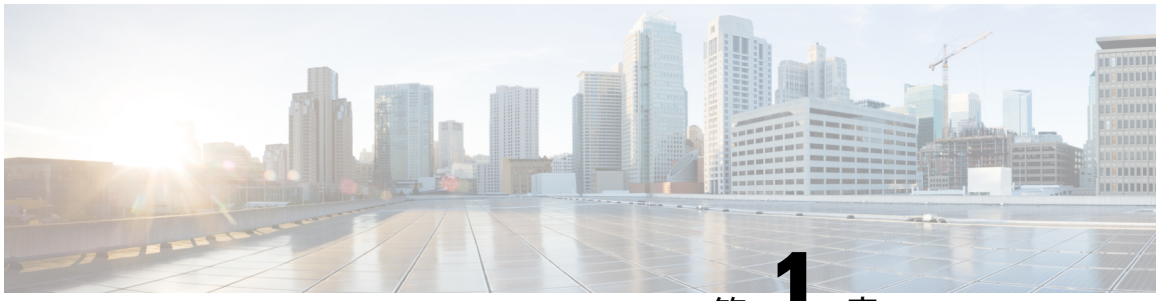
Cisco IOS XE Amsterdam 17.2.x（Catalyst 9200 スイッチ）インターフェイスおよびハードウェア コンポーネント コンフィギュレーション ガイド

初版：2020 年 3 月 30 日

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

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

目 次



第 1 章

インターフェイス特性の設定

インターフェイスの特性の概要

ここでは、インターフェイス特性について説明します。

インターフェイス タイプ

ここでは、デバイスでサポートされているインターフェイスのさまざまなタイプについて説明します。また、インターフェイスの物理特性に応じた設定手順についても説明します。

（注） このスタック対応の背面にあるスタックポートはイーサネットポートではないため設定できません。

ポートベースの VLAN

VLANは、ユーザの物理的な位置に関係なく、機能、チーム、またはアプリケーションなどで論理的に分割された、スイッチによるネットワークです。ポートで受信したパケットが転送されるのは、その受信ポートと同じVLANに属するポートに限られます。異なるVLAN上のネットワーク デバイスは、VLAN 間でトラフィックをルーティングするレイヤ 3 デバイスがなければ、互いに通信できません。

VLANに分割することにより、VLAN内でトラフィック用の堅固なファイアウォールを実現します。また、各 VLAN には固有の MAC アドレス テーブルがあります。VLAN が認識されるのは、ローカル ポートが VLAN に対応するように設定されたとき、VLAN Trunking Protocol

（VTP）トランク上のネイバーからその存在を学習したとき、またはユーザがVLANを作成したときです。スタック全体のポートを使用して VLAN を形成できます。

VLANを設定するには、**vlan** *vlan-id* 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、VLAN コンフィギュレーション モードを開始します。標準範囲 VLAN（VLAN ID 1 ～ 1005）の VLAN 設定は、VLAN データベースに保存されます。VTP がバージョン 1 または 2 の場合に、拡張範囲 VLAN（VLAN ID が 1006 ～ 4094）を設定するには、最初に VTP モードをトランスペアレントに設定する必要があります。トランスペアレント モードで作成された拡張範囲VLANは、VLANデータベースには追加されませんが、の実行コンフィギュレーションに保存されます。VTPバージョン3では、トランスペアレントモードの他に、クライアントモードまたはサーバモードで拡張範囲 VLAN を作成できます。これらの VLAN は VLAN データベースに格納されます。

スイッチ スタックでは、VLAN データベースはスタック内のすべてのスイッチにダウンロードされ、スタック内のすべてのスイッチによって同じ VLAN データベースが構築されます。スタックのすべてのスイッチで実行コンフィギュレーションおよび保存済みコンフィギュレーションが同一です。

インターフェイス コンフィギュレーション モードで **switchport** コマンドを使用すると、VLAN にポートが追加されます。

- インターフェイスを特定します。
- トランク ポートには、トランク特性を設定し、必要に応じて所属できる VLAN を定義します。
- アクセス ポートには、所属する VLAN を設定して定義します。

スイッチ ポート

スイッチ ポートは、物理ポートに対応付けられたレイヤ2専用インターフェイスです。スイッチ ポートは 1 つまたは複数の VLAN に所属します。スイッチ ポートは、アクセス ポートまたはトランク ポートにも使用できます。ポートは、アクセス ポートまたはトランク ポートに設定できます。また、ポート単位で Dynamic Trunking Protocol（DTP）を稼働させ、リンクのもう一端のポートとネゴシエートすることで、スイッチ ポート モードも設定できます。スイッチ ポートは物理インターフェイスおよび対応レイヤ 2 プロトコルの管理に使用します。ルーティングやブリッジングは処理しません。

スイッチポートの設定には、**switchport** インターフェイス コンフィギュレーション コマンドを使用します。

アクセス ポート

アクセス ポートは（音声 VLAN ポートとして設定されている場合を除き）1 つの VLAN だけに所属し、その VLAN のトラフィックだけを伝送します。トラフィックは、VLAN タグが付いていないネイティブ形式で送受信されます。アクセス ポートに着信したトラフィックは、ポートに割り当てられている VLAN に所属すると見なされます。アクセス ポートがタグ付きパケット（スイッチ間リンク（ISL）またはタグ付き IEEE 802.1Q）を受信した場合、そのパケットはドロップされ、送信元アドレスは学習されません。

サポートされているアクセス ポートのタイプは、次のとおりです。

- スタティック アクセス ポート。このポートは、手動でVLANに割り当てます（IEEE 802.1x で使用する場合は RADIUS サーバを使用します）。

また、Cisco IP Phone と接続するアクセス ポートを、1 つの VLAN は音声トラフィック用に、もう 1 つの VLAN は Cisco IP Phone に接続しているデバイスからのデータ トラフィック用に使用するように設定できます。

トランク ポート

トランク ポートは複数の VLAN のトラフィックを伝送し、デフォルトで VLAN データベース内のすべての VLAN のメンバとなります。次のトランク ポート タイプはサポートされています。

- ISL トランク ポートでは、受信パケットはすべて ISL ヘッダーを使用してカプセル化されているものと見なされ、送信パケットはすべて ISL ヘッダーとともに送信されます。ISL トランク ポートから受信したネイティブ（タグなし）フレームはドロップされます。
- IEEE 802.1Q トランク ポートは、タグ付きとタグなしの両方のトラフィックを同時にサポートします。IEEE 802.1Q トランク ポートは、デフォルトのポート VLAN ID（PVID）に割り当てられ、すべてのタグなしトラフィックはポートのデフォルト PVID 上を流れます。NULL VLAN ID を備えたすべてのタグなしおよびタグ付きトラフィックは、ポートのデフォルト PVID に所属するものと見なされます。発信ポートのデフォルト PVID と等しいVLAN ID を持つパケットは、タグなしで送信されます。残りのトラフィックはすべて、VLAN タグ付きで送信されます。

デフォルトでは、トランク ポートは、VTPに認識されているすべてのVLANのメンバですが、トランク ポートごとに VLAN の許可リストを設定して、VLAN メンバーシップを制限できます。許可 VLAN のリストは、その他のポートには影響を与えませんが、対応トランク ポートには影響を与えます。デフォルトでは、使用可能なすべての VLAN（VLAN ID 1 ～ 4094）が許可リストに含まれます。トランクポートは、VTP が VLAN を認識し、VLAN が有効な状態にある場合に限り、VLANのメンバーになることができます。VTPが新しい有効になっているVLAN を認識し、その VLAN がトランクポートの許可リストに登録されている場合、トランクポートは自動的にその VLAN のメンバになり、トラフィックはその VLAN のトランク ポート間で転送されます。VTPが、VLANのトランクポートの許可リストに登録されていない、新しい有効な VLAN を認識した場合、ポートはその VLAN のメンバーにはならず、その VLAN のトラフィックはそのポート間で転送されません。

トンネル ポート

トンネル ポートは IEEE 802.1Q トンネリングで使用され、サービスプロバイダー ネットワークのカスタマーのトラフィックを、同じ VLAN 番号を使用するその他のカスタマーから分離します。サービスプロバイダー エッジ スイッチのトンネル ポートからカスタマーのスイッチの IEEE 802.1Q トランク ポートに、非対称リンクを設定します。エッジ スイッチのトンネル ポートに入るパケットには、カスタマーのVLANですでにIEEE 802.1Qタグが付いており、カスタマーごとにIEEE 802.1Qタグの別のレイヤ（メトロ タグと呼ばれる）でカプセル化され、サービスプロバイダー ネットワークで一意の VLAN ID が含まれます。タグが二重に付いたパ

ケットは、その他のカスタマーのものとは異なる、元のカスタマーのVLANが維持されてサービスプロバイダー ネットワークを通過します。発信インターフェイス、およびトンネル ポートでは、メトロ タグが削除されてカスタマーのネットワークのオリジナル VLAN 番号が取得されます。

トンネル ポートは、トランク ポートまたはアクセス ポートにすることができず、それぞれのカスタマーに固有の VLAN に属する必要があります。

ルーテッド ポート

ルーテッドポートは物理ポートであり、ルータ上にあるポートのように動作しますが、ルータに接続されている必要はありません。ルーテッド ポートは、アクセス ポートとは異なり、特定の VLAN に対応付けられていません。VLAN サブインターフェイスをサポートしない点を除けば、通常のルータ インターフェイスのように動作します。 ルーテッド ポートは、レイヤ3ルーティング プロトコルで設定できます。ルーテッドポートはレイヤ3インターフェイス専用で、DTP や STP などのレイヤ 2 プロトコルはサポートしません。

ルーテッドポートを設定するには、**no switchport** インターフェイス コンフィギュレーション コマンドでインターフェイスをレイヤ 3 モードにします。次に、ポートに IP アドレスを割り当て、ルーティングを有効にして、**ip routing** および **router** *protocol* 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ルーティングプロトコルの特性を指定します。

（注） **no switchport**インターフェイス コンフィギュレーション コマンドを実行すると、インターフェイスがいったんシャットダウンされてから再度有効になり、インターフェイスが接続されているデバイスに関するメッセージが表示されることがあります。レイヤ2モードのインターフェイスをレイヤ3モードにした場合、影響のあるインターフェイスに関連する以前の設定が消失する可能性があります。

ソフトウェアに、設定できるルーテッドポートの個数制限はありません。ただし、ハードウェアには限界があるため、この個数と設定されている他の機能の数との相互関係によって CPU パフォーマンスに影響が及ぶことがあります。

スイッチ仮想インターフェイス

スイッチ仮想インターフェイス（SVI）は、スイッチポートの VLAN を、システムのルーティング機能に対する 1 つのインターフェイスとして表します。1 つの VLAN に関連付けることができる SVI は 1 つだけです。VLAN に対して SVI を設定するのは、VLAN 間でルーティングするため、またはデバイスにIPホスト接続を提供するためだけです。デフォルトでは、SVIはデフォルト VLAN（VLAN 1）用に作成され、リモートデバイスの管理を可能にします。追加の SVI は明示的に設定する必要があります。

（注） インターフェイス VLAN 1 は削除できません。

SVIはシステムにしかIPホスト接続を行いません。SVIは、VLANインターフェイスに対して**vlan**インターフェイス コンフィギュレーション コマンドを実行した際に初めて作成されます。

VLAN は、ISL または IEEE 802.1Q カプセル化トランク上のデータ フレームに関連付けられた VLAN タグ、あるいはアクセス ポート用に設定された VLAN ID に対応します。トラフィックをルーティングするそれぞれの VLAN に対して VLAN インターフェイスを設定し、IP アドレスを割り当ててください。

interface range コマンドを使用して、範囲内の既存の VLAN SVI を設定できます。interface range コマンド下で入力したコマンドは、範囲内の既存の VLAN SVI すべてに適用されます。コマンド **interface range create vlan** *x-y*を入力すると、まだ存在しない指定された範囲内のすべての vlan を作成できます。VLAN インターフェイスが作成されると、**interface range vlan** *id*を使用して VLAN インターフェイスを設定できます。

デバイススタックまたはスタンドアロンデバイスは合計 1,005 個の VLAN および SVI をサポートしますが、ハードウェアには限界があるため、SVI とルーテッド ポートの数および設定されている他の機能の数との相互関係によって、CPU パフォーマンスに影響が及ぶことがあります。

物理ポートと関連付けられていない場合、SVI を作成してもアクティブにはなりません。

EtherChannel ポートグループ

EtherChannel ポートグループは、複数のスイッチポートを 1 つのスイッチポートとして扱います。このようなポートグループは、デバイス間、またはデバイスとサーバ間で高帯域接続を行う単一論理ポートとして動作します。EtherChannel は、チャネルのリンク全体でトラフィックの負荷を分散させます。EtherChannel 内のリンクで障害が発生すると、それまでその障害リンクで伝送されていたトラフィックが残りのリンクに切り替えられます。複数のトランク ポートを 1 つの論理トランク ポートに、複数のアクセス ポートを 1 つの論理アクセス ポートに、複数のトンネル ポートを 1 つの論理トンネル ポートに、または複数のルーテッド ポートを 1 つの論理ルーテッドポートにグループ化できます。ほとんどのプロトコルは単一のまたは集約スイッチポートで動作し、ポート グループ内の物理ポートを認識しません。例外は、DTP、Cisco Discovery Protocol（CDP）、およびポート集約プロトコル（PAgP）で、物理ポート上でしか動作しません。

EtherChannel を設定するとき、ポートチャネル論理インターフェイスを作成し、EtherChannel にインターフェイスを割り当てます。レイヤ 3 インターフェイスの場合は、**interface port-channel** 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、論理インターフェイスを手動で作成します。その後、**channel-group** 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、インターフェイスを EtherChannel に手動で割り当てます。レイヤ 2 インターフェイスの場合は、**channel-group** 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、ポートチャネル論理インターフェイスを動的に作成します。このコマンドは物理および論理ポートをバインドします。

ネットワーク モジュール

次の表に、サポートされているアップリンクポートのリストを示します。

速度	C9200	C9200L
1 ギガビットイーサネット	—	固定アップリンク ポート
10 ギガビットイーサネット	モジュラアップリンク	固定アップリンク ポート

速度	C9200	C9200L
25 ギガビットイーサネット	モジュラアップリンク	固定アップリンク ポート

イーサネット接続が必要な場合は、すべてのモジュールの1ギガビットイーサネットにGLC-TE銅線 SFP を使用します。

次のSFP、SFP+、SFP28、QSFP ポートがサポートされています。

- 4x1G（C9200L のみ）
- 4x10G（C9200 と C9200L）
- 2x25G（C9200 と C9200L）

イーサネット経由の電力供給

Power over Ethernet（PoE）テクノロジーでは、PoE（802.3af標準規格）、PoE+（802.3at）ポートでデバイスの動作用の電源を供給できます。

詳細については、このガイドの「*PoE* の設定」の項を参照してください。

スイッチの USB ポートの使用

デバイスの前面パネルに 2 つの USB タイプ A ポートがあります。

USB ミニタイプ B コンソール ポート

デバイスには次のコンソールポートがあります。

- USB ミニタイプ B コンソール接続
- RJ-45 コンソール ポート

コンソール出力は両方のポートに接続されたデバイスに表示されますが、コンソール入力は一度に1つのポートしかアクティブになりません。デフォルトでは、USB コネクタはRJ-45 コネクタよりも優先されます。

（注） Windows PC には、USB ポートのドライバが必要です。ドライバ インストレーションの手順については、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。

付属の USB タイプ A ツー USB ミニタイプ B ケーブルを使用して PC または他のデバイスをこのデバイスを接続します。接続されたデバイスには、ターミナルエミュレーションアプリケーションが必要です。デバイスが、ホスト機能をサポートする電源の入っているデバイス（PC など）への有効な USB 接続を検出すると、RJ-45 コンソールからの入力がただちに無効になり、USB コンソールからの入力が有効になります。USB 接続が削除されると、RJ-45 コンソールからの入力はただちに再度有効になります。デバイスの LED はどの接続が使用中であるかを示します。

コンソール ポート変更ログ

ソフトウェア起動時に、ログに USB または RJ-45 コンソールのいずれがアクティブであるかが示されます。すべてのデバイスは常に RJ-45 メディアタイプを最初に表示します。

出力例では、デバイス 1 には接続された USB コンソールケーブルがあります。ブートローダが USB コンソールに変わらなかったため、デバイスからの最初のログは RJ-45 コンソールを示しています。少したってから、コンソールが変更され、USB コンソールログが表示されます。デバイス 2 とデバイス 3 には RJ-45 コンソールケーブルが接続されています。

```
switch-stack-1
*Mar  1 00:01:00.171: %USB_CONSOLE-6-MEDIA_RJ45: Console media-type is RJ45.
*Mar  1 00:01:00.431: %USB_CONSOLE-6-MEDIA_USB: Console media-type is USB.
```

USB ケーブルが取り外されるか、PC が USB 接続を非アクティブ化すると、ハードウェアは自動的に RJ-45 コンソール インターフェイスに変わります。

コンソール タイプが常に RJ-45 であるように設定でき、さらに USB コネクタの無活動タイムアウトを設定できます。

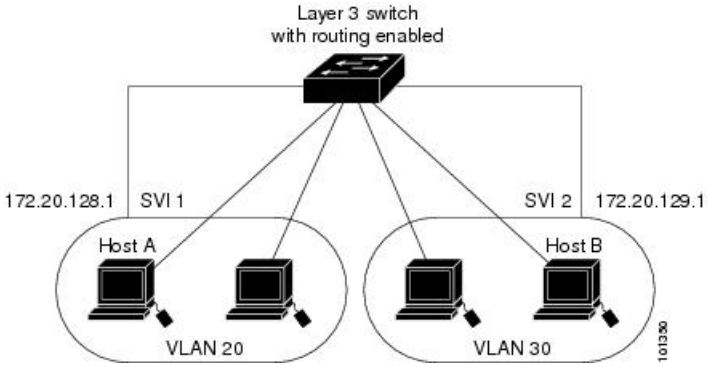
USB タイプ A ポート

USB タイプ A ポートは、外部 USB フラッシュ デバイス（サム ドライブまたは USB キーとも呼ばれる）へのアクセスを提供します。このポートは、容量 128 MB ～ 8 GB の Cisco USB フラッシュ ドライブをサポートします（ポート密度 128 MB、256 MB、1 GB、4 GB、8 GB の USB デバイスがサポートされます）。標準 Cisco IOS コマンドライン インターフェイス（CLI）コマンドを使用して、フラッシュ デバイスの読み取り、書き込み、および、コピー元やコピー先として使用できます。 を USB フラッシュドライブから起動するように設定することもできます。

インターフェイスの接続

単一 VLAN 内のデバイスは、スイッチを通じて直接通信できます。異なる VLAN に属すポート間では、ルーティングデバイスを介さなければデータを交換できません。標準のレイヤ 2 デバイスを使用すると、異なる VLAN のポートは、ルータを通じて情報を交換する必要があります。ルーティングが有効に設定されたデバイスの使用により、IP アドレスを割り当てた SVI で VLAN 20 および VLAN 30 の両方を設定すると、外部ルータを使用せずに、デバイスを介してホスト A からホスト B にパケットを直接送信できます。

図 *1:* スイッチと *VLAN* との接続



Network Advantage ライセンスがデバイスまたはアクティブなデバイスで使用されている場合は、そのデバイスがルーティング方式を使用してインターフェイス間のトラフィックを転送します。Network Essentials ライセンスがデバイスまたはアクティブなデバイスで使用されている場合は、基本ルーティング（静的ルーティングと RIP）だけがサポートされます。可能な場合は、高いパフォーマンスを維持するために、転送はデバイスハードウェアで実行されます。ただし、ハードウェアでルーティングされるのはイーサネット II カプセル化された IPv4 パケットだけです。

ルーティング機能は、すべての SVI およびルーテッド ポートで有効にできます。デバイスは IP トラフィックだけをルーティングします。IP ルーティング プロトコル パラメータとアドレス設定が SVI またはルーテッド ポートに追加されると、このポートで受信した IP トラフィックはルーティングされます。

インターフェイス コンフィギュレーション モード

デバイスは、次のインターフェイスタイプをサポートします。

- 物理ポート：デバイスポートおよびルーテッドポート
- VLAN：スイッチ仮想インターフェイス
- ポートチャネル：EtherChannel インターフェイス

インターフェイス範囲も設定できます。

物理インターフェイス（ポート）を設定するには、インターフェイスタイプ、スタック メンバー番号（スタッキング対応スイッチのみ）、モジュール番号、およびデバイスのポート番号を指定し、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。

- タイプ：10/100/1000 Mbps イーサネットポートの場合はギガビットイーサネット（GigabitEthernet または gi）、10 Gbps の場合は 10 ギガビットイーサネット（TenGigabitEthernet または te）、25 Gbps の場合は 25 ギガビットイーサネット（TwentyFiveGigE or twe）、40 Gbps の場合は Small Form-Factor Pluggable（SFP）モジュール ギガビット イーサネット インターフェイス。

- スイッチポート LED をスタックモードで使用して、デバイスのスタックメンバー番号を識別できます。
- モジュール番号：デバイス上のモジュールまたはスロット番号：スイッチ（ダウンリンク）ポートは 0 で、アップリンクポートは 1 です。
- SFP アップリンクポートを装着したデバイスの場合、モジュール番号は 1 で、ポート番号が振り直されます。たとえば、デバイスに 10/100/1000 ポートが 24 個ある場合、SFP モジュールポートは、GigabitEthernet1/1/1 ～ GigabitEthernet1/1/4、または TenGigabitEthernet1/1/1 ～ TenGigabitEthernet1/1/4 になります。

デバイス上のインターフェイスの位置を物理的に確認することで、物理インターフェイスを識別できます。**show** 特権 EXEC コマンドを使用して、スイッチ上の特定のインターフェイスまたはすべてのインターフェイスに関する情報を表示することもできます。以降、この章では、主に物理インターフェイスの設定手順について説明します。

次に、スタッキング対応およびスタンドアロンデバイスでインターフェイスを設定する例を示します。

- スタンドアロンデバイスで 10/100/1000 ポート 4 を設定するには、次のコマンドを入力します。

```
Device# configure terminal
Device(config)# interface GigabitEthernet1/0/4
```

- スタンドアロンデバイスで 10 ギガビットイーサネット ポート 1 を設定するには、次のコマンドを入力します。

```
Device# configure terminal
Device(config)# interface TenGigabitEthernet 1/1/1
```

- スタック メンバー 3 に 10 ギガビットイーサネット ポートを設定するには、次のコマンドを入力します。

```
Device# configure terminal
Device(config)# interface TenGigabitEthernet 3/1/1
```

- スタンドアロンデバイスで最初の SFP モジュール（アップリンク）を設定するには、次のコマンドを入力します。

```
Device# configure terminal
Device(config)# interface GigabitEthernet 1/1/1
```

イーサネット インターフェイスのデフォルト設定

インターフェイスがレイヤ3モードの場合に、レイヤ2パラメータを設定するには、パラメータを指定せずに **switchport** 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、インターフェイスをレイヤ 2 モードにする必要があります。これにより、インターフェイスがいったんシャットダウンしてから再度有効になり、インターフェイスが接続しているデバイスに関するメッセージが表示されることがあります。レイヤ3モードのインターフェイスをレイヤ2モードにした場合、影響のあるインターフェイスに関連する以前の設定情報が消失する可能性があり、インターフェイスはデフォルト設定に戻ります。

次の表は、レイヤ2インターフェイスにのみ適用される一部の機能を含む、イーサネットインターフェイスのデフォルト設定を示しています。

表 *1:* レイヤ *2* イーサネット インターフェイスのデフォルト設定

機能	デフォルト設定
動作モード	レイヤ 2 またはスイッチングモード（**switchport** コマンド）。
VLAN 許容範囲	VLAN 1 ～ 4094
デフォルト VLAN（アクセス ポート用）	VLAN 1（レイヤ 2 インターフェイスだけ）。
ネイティブ VLAN（IEEE 802.1Q トランク用）	VLAN 1（レイヤ 2 インターフェイスだけ）。
VLAN トランキング	Switchport mode dynamic auto（DTP をサポート）（レイヤ 2 インターフェイスだけ）。
ポート イネーブル ステート	すべてのポートが有効。
ポート記述	未定義。
速度	自動ネゴシエーション
デュプレックス モード	自動ネゴシエーション
フロー制御	フロー制御は **receive: off** に設定されます。送信パケットでは常にオフです。
EtherChannel（PAgP）	すべてのイーサネット ポートで無効。
ポート ブロッキング（不明マルチキャストおよび不明ユニキャスト トラフィック）	無効（ブロッキングされない）（レイヤ2インターフェイスだけ）。
ブロードキャスト、マルチキャスト、およびユニキャスト ストーム制御	無効。

機能	デフォルト設定
保護ポート	無効（レイヤ 2 インターフェイスだけ）。
ポート セキュリティ	無効（レイヤ 2 インターフェイスだけ）。
PortFast	無効。
Auto-MDIX	有効。 （注） IEEE 802.3af に完全には準拠していない Cisco IP 電話やアクセスポイントなど、準規格の受電デバイスについては、その受電デバイスをクロスケーブルでスイッチに接続する場合、スイッチでサポートされないことがあります。これは、スイッチ ポート上で Automatic Medium-Dependent Interface Crossover（Auto-MIDX）が有効かどうかは関係ありません。
Power over Ethernet（PoE）	有効（auto）。

インターフェイス速度およびデュプレックス モード

スイッチのギガビットイーサネットのインターフェイスは、10 Mbps、100 Mbps、1000 Mbps のいずれかの速度で、かつ全二重か半二重のどちらかのモードで動作します。全二重モードの場合、2 つのステーションが同時にトラフィックを送受信できます。通常、10 Mbps ポートは半二重モードで動作します。つまり、ステーションはトラフィックの受信または送信のいずれかを交互に行います。また、スイッチには 100 Mb、1 Gb、2.5 Gb、5 Gb、および 10 Gb の速度をサポートし、全二重モードで動作するマルチギガビット イーサネット ポート（最大 1 Gbps の速度をサポートする SFP モジュール、最大 10 Gbps の速度をサポートする SFP+ モジュール、最大 25 Gbps をサポートする SFP28 モジュール、および最大 40 Gbps の速度をサポートする QSFP）が搭載されています。サポートされているスイッチモデルのリストについては、『*Cisco Catalyst 9200 Series Switches Hardware Installation Guide*』を参照してください。

速度とデュプレックス モードの設定時の注意事項

インターフェイス速度とデュプレックスモードを設定する際には、次のガイドラインに注意してください。

- ギガビットイーサネット（10/100/1000 Mbps）ポートはすべての速度オプションとすべてのデュプレクスオプション（自動、半二重、および全二重）をサポートします。ただし、1000 Mbps 以上で動作するギガビット イーサネット ポートは半二重モードをサポートしません。

マルチギガビット イーサネット ポート（100 Mbps、1 Gbps、2.5 Gbps、5 Gbps、10 Gbps、100 Gbps）はすべての速度オプションをサポートし、自動および全二重モードのみをサポートします。これらのポートはどの速度でも半二重モードをサポートしません。

1 Gbps で動作している SFP ポート、10 Gbps で動作している SFP+ ポート、25 Gbps で動作している SFP28 ポートおよび 40 Gbps で動作している QSFPポートは **no speed nonegotiate** または **speed nonegotiate** です。デュプレックス オプションはサポートされません。

（注） SFP、SFP+、および SFP28 ポートは、1000Base-T SFP が使用されている場合にのみ、速度（自動、10、100、1000）およびデュプレックス（自動/全二重/半二重）オプションをサポートします。SFP、SFP+、および SFP28 ポートは、GLC-GE-100FX モジュールが使用されている場合にのみ、速度（自動/100）およびデュプレックス（自動/全二重/半二重）オプションをサポートします。

- 回線の両側で自動ネゴシエーションがサポートされる場合は、デフォルト設定の **auto** ネゴシエーションの使用を強くお勧めします。
- 一方のインターフェイスが自動ネゴシエーションをサポートし、もう一方がサポートしない場合は、両方のインターフェイス上でデュプレックスと速度を設定します。サポートする側で **auto** 設定を使用しないでください。
- STP が有効な場合にポートを再設定すると、デバイスがループの有無を調べるために最大で 30 秒かかる可能性があります。STP の再設定が行われている間、ポート LED はオレンジに点灯します。ベストプラクティスとして、速度とデュプレックスのオプションをリンク上で **auto** に設定するか、リンク終端の両側で **fixed** に設定することを推奨します。リンクの片側を **auto** に設定し、もう一方を **fixed** に設定した場合はリンクが起動しませんが、これは予期される動作です。
- ベスト プラクティスとして、速度とデュプレックスのオプションをリンク上で **auto** に設定するか、リンク終端の両側で **fixed** に設定することを推奨します。リンクの片側を **auto** に設定し、もう一方を **fixed** に設定した場合はリンクが起動しませんが、これは予期される動作です。

注意 インターフェイス速度とデュプレックス モードの設定を変更すると、再設定中にインターフェイスがシャットダウンし、再び有効になる場合があります。

IEEE 802.3x フロー制御

フロー制御により、接続しているイーサネット ポートは、輻輳しているノードがリンク動作をもう一方の端で一時停止できるようにすることによって、輻輳時のトラフィック レートを制御できます。あるポートで輻輳が生じ、それ以上はトラフィックを受信できなくなった場合、ポーズフレームを送信することによって、その状態が解消されるまで送信を中止するように、

そのポートから相手ポートに通知します。ポーズ フレームを受信すると、送信側デバイスはデータ パケットの送信を中止するので、輻輳時のデータ パケット損失が防止されます。

（注） スイッチ ポートは、ポーズ フレームを受信できますが、送信はできません。

flowcontrol 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インターフェイスのポーズフレームを **receive** する機能を **on**、**off**、、または **desired** に設定できます。デフォルトの状態は **on** です。

desired に設定した場合、インターフェイスはフロー制御パケットの送信を必要とする接続デバイスか、または必要ではないもののフロー制御パケットを送信できる接続デバイスで動作できます。

デバイスのフロー制御設定には、次のルールが適用されます。

- **receive on**（または **desired**）：ポートはポーズフレームを送信できませんが、ポーズフレームを送信する必要のある、または送信できる接続デバイスと組み合わせて使用できます。ポーズフレームの受信は可能です。
- **receive off**：フロー制御はどちらの方向にも動作しません。輻輳が生じても、リンクの相手側に通知はなく、どちら側の装置も休止フレームの送受信を行いません。

（注） コマンドの設定と、その結果生じるローカルおよびリモートポートでのフロー制御解決の詳細については、このリリースのコマンドリファレンスに記載された **flowcontrol** インターフェイス コンフィギュレーション コマンドを参照してください。

レイヤ 3 インターフェイス

デバイスは、次のレイヤ 3 インターフェイスをサポートします。

- SVI：トラフィックをルーティングする VLAN に対応する SVI を設定する必要があります。SVI は、**interface vlan** グローバル コンフィギュレーション コマンドのあとに VLAN ID を入力して作成します。SVI を削除するには、**no interface vlan** 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。インターフェイス VLAN 1 は削除できません。

  （注） 物理ポートと関連付けられていない場合、SVI を作成してもアクティブにはなりません。

  SVI を設定するとき、ポートで **switchport autostate exclude** コマンドを使用して、SVI ラインステートを判断する際に含めないようにできます。SVI で自動ステートを無効にするには、SVI で **no autostate** コマンドを使用します。

- ルーテッドポート：ルーテッドポートは、**no switchport** 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、レイヤ 3 モードになるように設定された物理ポートです。ルーテッドポートは VLAN サブインターフェイスをサポートします。

  VLANサブインターフェイス：802.1Q VLANサブインターフェイスは、ルーテッド物理インターフェイス上のVLAN IDに関連付けられた仮想Cisco IOSインターフェイスです。親インターフェイスは物理ボートです。サブインターフェイスはレイヤ3物理インターフェイス上にのみ作成できます。サブインターフェイスは、IPアドレッシング、転送ポリシー、Quality of Service（QoS）ポリシー、セキュリティポリシーなどのさまざまな機能に関連付けることができます。親インターフェイスはサブインターフェイスによって複数の仮想インターフェイスに分割されます。これらの仮想インターフェイスに IP アドレスやダイナミック ルーティング プロトコルなど固有のレイヤ 3 パラメータを割り当てることができます。各サブインターフェイスのIPアドレスは、親インターフェイスの他のサブインターフェイスのサブネットとは異なります。

- レイヤ 3 EtherChannel ポート：EtherChannel インターフェイスは、ルーテッド ポートで構成されます。

レイヤ 3 デバイスは、各ルーテッドポートおよび SVI に割り当てられた IP アドレスを持つことができます。

デバイスまたはデバイススタックで設定可能なSVIとルーテッドポートの数に対して定義された制限はありません。ただし、ハードウェアには限界があるため、SVIおよびルーテッドポートの個数と、設定されている他の機能の個数の組み合わせによっては、CPU利用率が影響を受けることがあります。デバイスが最大限のハードウェアリソースを使用している場合にルーテッドポートまたは SVI を作成しようとすると、次のような結果になります。

- 新たなルーテッドポートを作成しようとすると、デバイスはインターフェイスをルーテッドポートに変換するための十分なリソースがないことを示すメッセージを表示し、インターフェイスはスイッチポートのままとなります。

- 拡張範囲の VLAN を作成しようとすると、エラー メッセージが生成され、拡張範囲の VLAN は拒否されます。

- VLAN Trunking Protocol（VTP）が新たな VLAN をデバイスに通知すると、使用可能な十分なハードウェアリソースがないことを示すメッセージを送り、その VLAN をシャットダウンします。**show vlan** EXEC コマンドの出力に、中断状態の VLAN が示されます。

- デバイスが、ハードウェアのサポート可能な数を超える VLAN とルーテッドポートが設定されたコンフィギュレーションを使って起動を試みると、VLAN は作成されますが、ルーテッドポートはシャットダウンされ、デバイスはハードウェアリソースが不十分であるという理由を示すメッセージを送信します。

（注） すべてのレイヤ 3 インターフェイスには、トラフィックをルーティングするための IP アドレスが必要です。次の手順は、レイヤ3インターフェイスとしてインターフェイスを設定する方法およびインターフェイスに IP アドレスを割り当てる方法を示します。

物理ポートがレイヤ 2 モードである（デフォルト）場合は、**no switchport** インターフェイス コンフィギュレーション コマンドを実行してインターフェイスをレイヤ3モードにする必要があります。**no switchport** コマンドを実行すると、インターフェイスが無効化されてから再度有効になります。これにより、インターフェイスが接続しているデバイスに関するメッセージが生成されることがあります。さらに、レイヤ2モードのインターフェイスをレイヤ3モードにすると、影響を受けたインターフェイスに関連する前の設定情報は失われ、インターフェイスはデフォルト設定に戻る可能性があります。

インターフェイス特性の設定方法

次の項では、インターフェイス特性を設定する手順を構成するさまざまなタスクについて説明します。

インターフェイスの設定

次の一般的な手順は、すべてのインターフェイス設定プロセスに当てはまります。

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	**interface** 例：	インターフェイスタイプ、 およびコネクタの数を識別します。

	コマンドまたはアクション	目的
	`Device(config)# interface gigabitethernet1/0/1` `Device(config-if)#`	（注） インターフェイス タイプとインターフェイス番号の間にスペースを入れる必要はありません。たとえば、前の行では、**gigabitethernet 1/0/1**、**gigabitethernet1/0/1**、**gi 1/0/1**、または **gi1/0/1** のいずれかを指定できます。
ステップ 4	各 **interface** コマンドの後ろに、インターフェイスに必要なインターフェイス コンフィギュレーション コマンドを続けて入力します。	インターフェイス上で実行するプロトコルとアプリケーションを定義します。別のインターフェイス コマンドまたは **end** を入力して特権 EXEC モードに戻ると、コマンドが収集されてインターフェイスに適用されます。
ステップ 5	**interface range** または **interface range macro**	（任意）インターフェイスの範囲を設定します。 （注） ある範囲内で設定したインターフェイスは、同じタイプである必要があります。また、同じ機能オプションを指定して設定しなければなりません。
ステップ 6	**show interfaces**	スイッチ上のまたはスイッチに対して設定されたすべてのインターフェイスのリストを表示します。デバイスがサポートする各インターフェイスまたは指定したインターフェイスのレポートが出力されます。

インターフェイスに関する記述の追加

インターフェイスの記述を追加するには、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：	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
	`Device> enable`	
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-id* 例： `Device(config)# interface gigabitethernet1/0/2`	記述を追加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 4	**description** *string* 例： `Device(config-if)# description Connects to Marketing`	インターフェイスに記述を追加します。
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 interfaces** *interface-id* **description**	入力を確認します。
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

インターフェイス範囲の設定

同じ設定パラメータを持つ複数のインターフェイスを設定するには、**interface range** 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。インターフェイス レンジ コンフィギュレーション モードを開始すると、このモードを終了するまで、入力されたすべてのコマンド パラメータはその範囲内のすべてのインターフェイスに対するものと見なされます。

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	**interface range** {*port-range* | **macro** *macro_name*} 例： `Device(config)# interface range macro`	設定するインターフェイス範囲（VLAN または物理ポート）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。 • **interface range** コマンドを使用すると、最大 5 つのポート範囲または定義済みマクロを 1 つ設定できます。 • **macro** 変数は、「インターフェイス レンジ マクロの設定および使用方法」で説明されています。 • カンマで区切った *port-range* では、各エントリに対応するインターフェイス タイプを入力し、カンマの前後にスペースを含めます。 • ハイフンで区切った *port-range* では、インターフェイス タイプの再入力は不要ですが、ハイフンの前後にスペースを入力する必要があります。 （注） この時点で、通常のコンフィギュレーション コマンドを使用して、範囲内のすべてのインターフェイスにコンフィギュレーション パラメータを適用します。各コマンドは、入力されたとおりに実行されます。

	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。
ステップ 5	**show interfaces** [*interface-id*] 例： `Device# show interfaces`	指定した範囲内のインターフェイスの設定を確認します。
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。

インターフェイス レンジ マクロの設定および使用方法

インターフェイス レンジ マクロを作成すると、設定するインターフェイスの範囲を自動的に選択できます。**interface range macro** グローバル コンフィギュレーション コマンド文字列で **macro** キーワードを使用する前に、**define interface-range** 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マクロを定義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	**define interface-range** *macro_name interface-range* 例：	インターフェイス範囲マクロを定義して、NVRAM に保存します。 • *macro_name* は、最大 32 文字の文字列です。

	コマンドまたはアクション	目的
		• マクロには、カンマで区切ったインターフェイスを5つまで指定できます。 • それぞれの *interface-range* は、同じポート タイプで構成されていなければなりません。 （注） **interface range macro** グローバル コンフィギュレーション コマンド文字列で **macro** キーワードを使用する前に、**define interface-range** 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マクロを定義する必要があります。
ステップ 4	**interface range macro** *macro_name* 例： `Device(config)# interface range macro enet_list`	*macro_name* の名前でインターフェイス範囲マクロに保存された値を使用することによって、設定するインターフェイスの範囲を選択します。 ここで、通常のコンフィギュレーション コマンドを使用して、定義したマクロ内のすべてのインターフェイスに設定を適用できます。
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 | include define** 例： `Device# show running-config | include define`	定義済みのインターフェイス範囲マクロの設定を表示します。
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

インターフェイス速度およびデュプレックス パラメータの設定

インターフェイスの速度とデュプレックスパラメータを設定するには、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。 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-id* 例： `Device(config)# interface gigabitethernet1/0/3`	設定する物理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 4	**duplex** {**auto** | **full** | **half**} 例： `Device(config-if)# duplex half`	インターフェイスのデュプレックス パラメータを入力します。 半二重モードを有効にします（10 Mb/s または 100 Mb/s のみで動作するインターフェイスの場合）。半二重は、1000 Mb/s の速度に設定されたマルチギガビット イーサネット ポートではサポートされていません。 デュプレックス設定を行うことができるのは、速度が **auto** に設定されている場合です。
ステップ 5	**end** 例： `Device(config-if)# 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 6	**show interfaces** *interface-id* 例： `Device# show interfaces gigabitethernet1/0/3`	インターフェイス速度およびデュプレックス モードの設定を表示します。
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

IEEE 802.3x フロー制御の設定

IEEE 802.3x フロー制御を設定するには、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。 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-id* 例： `Device(config)# interface gigabitethernet1/0/1`	設定する物理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 4	**flowcontrol** {**receive**} {**on** | **off** | **desired**} 例： `Device(config-if)# flowcontrol receive on`	ポートのフロー制御モードを設定します。

	コマンドまたはアクション	目的
ステップ 5	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show interfaces** *interface-id* 例： `Device# show interfaces gigabitethernet1/0/1`	インターフェイス フロー制御の設定を確認します。
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

レイヤ 3 インターフェイスの設定

レイヤ 3 インターフェイスを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 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** { **gigabitethernet** *interface-id*} | { **vlan** *vlan-id*} | { **port-channel** *port-channel-number*} 例： `Device(config)# interface`	レイヤ3インターフェイスとして設定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。

	コマンドまたはアクション	目的
	gigabitethernet1/0/2	
ステップ 4	**no switchport** 例： `Device(config-if)# no switchport`	（物理ポートの場合のみ）レイヤ 3 モードを開始します。
ステップ 5	**ip address** *ip_address subnet_mask* 例： `Device(config-if)# ip address 192.20.135.21 255.255.255.0`	IP アドレスおよび IP サブネットを設定します。
ステップ 6	**no shutdown** 例： `Device(config-if)# no shutdown`	インターフェイスを有効にします。
ステップ 7	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。
ステップ 8	**show interfaces** [*interface-id*]	設定を確認します。
ステップ 9	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

論理レイヤ 3 GRE トンネルインターフェイスの設定

始める前に

総称ルーティング カプセル化（GRE）は、仮想ポイントツーポイント リンク内でネットワーク層プロトコルをカプセル化するために使用されるトンネリング プロトコルです。GRE トンネルは、カプセル化のみを提供し、暗号化は提供しません。

（注）

- GRE トンネルは、Cisco Catalyst 9000 スイッチのハードウェアでサポートされています。GRE でトンネル オプションを設定しない場合、パケットはハードウェアでスイッチングされます。GREをトンネルオプション（キーやチェックサムなど）で設定すると、パケットはソフトウェアでスイッチングされます。最大 10 つの GRE トンネルがサポートされます。
- GRE トンネルではアクセスコントロールリスト（ACL）や Quality of Service（QoS）などのその他の機能はサポートされません。
- GREトンネルでは **tunnel path-mtu-discovery** コマンドはサポートされていません。フラグメンテーションを回避するには、**ip mtu 256** コマンドを使用して GRE トンネルの両端の最大伝送ユニット（MTU）を最小値に設定します。

GRE トンネルを設定する手順は、次のとおりです。

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	**interface tunnel** *number* 例： `Device(config)#interface tunnel 2`	インターフェイスでトンネリングを有効にします。
ステップ 4	**ip address** *ip_addresssubnet_mask* 例： `Device(config)#ip address 100.1.1.1 255.255.255.0`	IP アドレスおよび IP サブネットを設定します。
ステップ 5	**tunnel source** {*ip_address* | *type_number*} 例： `Device(config)#tunnel source 10.10.10.1`	トンネル送信元を設定します。
ステップ 6	**tunnel destination** {*host_name* | *ip_address*}	トンネル宛先を設定します。

	コマンドまたはアクション	目的
	例： `Device(config)#tunnel destination 10.10.10.2`	
ステップ 7	**tunnel mode gre ip** 例： `Device(config)#tunnel mode gre ip`	トンネル モードを設定します。
ステップ 8	**end** 例： `Device(config)#end`	設定モードを終了します。

SVI 自動ステート除外の設定

SVI 自動ステートを除外するには、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	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet1/0/2`	レイヤ2インターフェイス（物理ポートまたはポート チャネル）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 4	**switchport autostate exclude** 例： `Device(config-if)# switchport autostate exclude`	SVI ライン ステート（アップまたはダウン）のステータスを定義する際、アクセスまたはトランク ポートを除外します。

	コマンドまたはアクション	目的
ステップ 5	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show running config interface** *interface-id*	（任意）実行コンフィギュレーションを表示します。 設定を確認します。
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

インターフェイスのシャットダウンおよび再起動

インターフェイスをシャットダウンすると、指定されたインターフェイスのすべての機能が無効になり、使用不可能であることがすべてのモニタ コマンドの出力に表示されます。この情報は、すべてのダイナミック ルーティング プロトコルを通じて、他のネットワーク サーバに伝達されます。ルーティング アップデートには、インターフェイス情報は含まれません。

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	**interface** { **vlan** *vlan-id*} | { **gigabitethernet** *interface-id*} | { **port-channel** *port-channel-number*} 例： `Device(config)# interface`	設定するインターフェイスを選択します。

	コマンドまたはアクション	目的
	gigabitethernet1/0/2	
ステップ 4	**shutdown** 例： `Device(config-if)# shutdown`	インターフェイスをシャットダウンします。
ステップ 5	**no shutdown** 例： `Device(config-if)# no shutdown`	インターフェイスを再起動します。
ステップ 6	**end** 例： `Device(config-if)# 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。

コンソール メディア タイプの設定

コンソール メディア タイプを RJ-45 に設定するには、次の手順を実行します。RJ-45 としてコンソールを設定すると、USB コンソールの動作は無効になり、入力は RJ-45 コネクタからのみ供給されます。

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：	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
	`Device# configure terminal`	
ステップ 3	**line console 0** 例： `Device(config)# line console 0`	コンソールを設定し、ライン 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	**media-type rj45 switch** *switch_number* 例： `Device(config-line)# media-type rj45 switch 1`	コンソール メディア タイプが RJ-45 ポート以外に設定されないようにします。このコマンドを入力せず、両方のタイプが接続された場合は、デフォルトで USB ポートが使用されます。
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	**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。

USB 無活動タイムアウトの設定

タイムアウトのために USB コンソール ポートは非アクティブ化された場合、USB ポートを切断し、再接続すると、動作を回復できます。

（注） 設定された無活動タイムアウトはスタックのすべてのデバイスに適用されます。ただし、あるデバイスのタイムアウトによってスタック内の別のデバイスがタイムアウトを引き起こすことはありません。

手順

	コマンドまたはアクション	目的
ステップ 1	**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	**line console 0** 例： `Device(config)# line console 0`	コンソールを設定し、ライン 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	**usb-inactivity-timeout switch** *switch_number timeout-minutes* 例： `Device(config-line)# usb-inactivity-timeout switch 1 30`	コンソールポートの無活動タイムアウトを指定します。指定できる範囲は 1 ～ 240 分です。デフォルトでは、タイムアウトが設定されていません。
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。

インターフェイス特性のモニタ

ここでは、インターフェイス特性のモニタリングについて説明します。

インターフェイス ステータスの監視

特権 EXEC プロンプトにコマンドを入力することによって、ソフトウェアおよびハードウェアのバージョン、コンフィギュレーション、インターフェイスに関する統計情報などのインターフェイス情報を表示できます。

表 *2:* インターフェイス用の *show* コマンド

コマンド	目的
show interfaces *interface-id* **status** [**err-disabled**]	インターフェイスのステータスまたは error-disabled ステートにあるインターフェイスのリストを表示します。

コマンド	目的
show interfaces [*interface-id*] **switchport**	スイッチング（非ルーティング）ポートの管理上および動作上のステータスを表示します。このコマンドを使用すると、ポートがルーティングまたはスイッチングのどちらのモードにあるかが判別できます。
show interfaces [*interface-id*] **description**	1つのインターフェイスまたはすべてのインターフェイスに関する記述とインターフェイスのステータスを表示します。
show ip interface [*interface-id*]	IP ルーティング用に設定されたすべてのインターフェイスまたは特定のインターフェイスについて、使用できるかどうかを表示します。
show interface [*interface-id*] **stats**	インターフェイスのパスごとに入出力パケットを表示します。
show interfaces *interface-id*	（任意）インターフェイスの速度およびデュプレックスを表示します。
show interfaces transceiver dom-supported-list	（任意）接続 SFP モジュールの Digital Optical Monitoring（DOM）ステータスを表示します。
show interfaces transceiver properties	（任意）インターフェイスの温度、電圧、電流量を表示します。
show interfaces [*interface-id*] [{**transceiver properties** | **detail**}] *module number*]	SFP モジュールに関する物理および動作ステータスを表示します。
show running-config interface [*interface-id*]	インターフェイスに対応する RAM 上の実行コンフィギュレーションを表示します。
show version	ハードウェア設定、ソフトウェア バージョン、コンフィギュレーション ファイルの名前と送信元、およびブートイメージを表示します。
show controllers ethernet-controller *interface-id* **phy**	インターフェイスの Auto-MDIX 動作ステートを表示します。

インターフェイスおよびカウンタのクリアとリセット

表 *3:* インターフェイスの *clear* コマンド

コマンド	目的
clear counters [*interface-id*]	インターフェイス カウンタをクリアします。

コマンド	目的
clear interface *interface-id*	インターフェイスのハードウェア ロジックをリセットします。
clear line [*number* | **console 0** | **vty** *number*]	非同期シリアル回線に関するハードウェア ロジックをリセットします。

（注） **clear counters** 特権 EXEC コマンドは、簡易ネットワーク管理プロトコル（SNMP）を使用して取得されたカウンタをクリアしません。**show interface** 特権 EXEC コマンドで表示されるカウンタのみをクリアします。

インターフェイス特性の設定例

この項では、インターフェイス特性の設定例を示します。

例：インターフェイスの説明の追加

次に、インターフェイスの説明を追加する例を示します。

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTRL/Z.
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# description Connects to Marketing
Device(config-if)# end
Device# show interfaces gigabitethernet1/0/2 description
Interface Status         Protocol Description
Gi1/0/2   admin down     down     Connects to Marketing
```

例：スタック対応スイッチでのインターフェイスの設定

次に、スタンドアロンスイッチ上で 10/100/1000 ポート 4 を設定する例を示します。

```
Device(config)# interface gigabitethernet1/1/4
```

次に、スタックメンバー 1 で最初の SFP モジュールのアップリンクポートを設定する例を示します。

```
Device(config)# interface gigabitethernet1/1/1
```

次に、スタックメンバー 3 で10 ギガビット イーサネット ポートを設定する例を示します。

```
Device(config)# interface tengigabitethernet3/0/1
```

例：インターフェイスの範囲の設定

次に、**interface range** 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、スイッチ 1 のポート 1 ～ 4 で速度を 100 Mb/s に設定する例を示します。

```
Device# configure terminal
Device(config)# interface range gigabitethernet1/0/1 - 4
Device(config-if-range)# speed 100
```

次に、カンマを使用して異なるインターフェイス タイプ ストリングを範囲に追加し、ギガビット イーサネット ポート 1 ～ 3 と、10 ギガビット イーサネット ポート 1 および 2 の両方を有効にし、フロー制御ポーズ フレームを受信する例を示します。

```
Device# configure terminal
Device(config)# interface range gigabitethernet1/1/1 - 3 , tengigabitethernet1/1/1 - 2
Device(config-if-range)# flowcontrol receive on
```

（注） インターフェイス レンジ モードで複数のコンフィギュレーション コマンドを入力した場合、各コマンドは入力した時点で実行されます。インターフェイス レンジモードを終了した後で、コマンドがバッチ処理されるわけではありません。コマンドの実行中にインターフェイス レンジコンフィギュレーションモードを終了すると、一部のコマンドが範囲内のすべてのインターフェイスに対して実行されない場合もあります。コマンドプロンプトが再表示されるのを待ってから、インターフェイス範囲コンフィギュレーション モードを終了してください。

例：インターフェイス範囲のマクロ設定と使用方法

次に、インターフェイス範囲のマクロ *enet_list* に対するインターフェイス レンジ コンフィギュレーション モードを開始する例を示します。

```
Device# configure terminal
Device(config)# interface range macro enet_list
Device(config-if-range)#
```

次に、インターフェイス範囲のマクロ *enet_list* を削除し、処理を確認する例を示します。

```
Device# configure terminal
Device(config)# no define interface-range enet_list
Device(config)# end
Device# show run | include define
Device#
```

例：インターフェイス速度とデュプレックスモードの設定

次に、10/100/1000 Mbps ポートでインターフェイス速度を 10 Mbps、デュプレクスモードを全二重にする例を示します。

```
Device# configure terminal
Device(config)# interface gigabitethernet1/0/3
Device(config-if)# speed 10
Device(config-if)# duplex full
```

次に、10/100/1000 Mbps ポートでインターフェイス速度を 100 Mbps に設定する例を示します。

```
Device# configure terminal
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# speed 100
```

例：レイヤ 3 インターフェイスの設定

次に、レイヤ 3 インターフェイスを設定する例を示します。

```
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# no switchport
Device(config-if)# ip address 192.20.135.21 255.255.255.0
Device(config-if)# no shutdown
```

例：コンソールメディアタイプの設定

次に、USB コンソールメディアタイプを無効にし、RJ-45 コンソールメディアタイプを有効にする例を示します。

```
Device# configure terminal
Device(config)# line console 0
Device(config-line)# media-type rj45 switch 1
```

この設定は、スタック内のすべてのアクティブな USB コンソール メディア タイプを終了します。ログにはこの終了の発生が示されます。次に、スイッチ 1 のコンソールが RJ-45 に戻る例を示します。

```
*Mar  1 00:25:36.860: %USB_CONSOLE-6-CONFIG_DISABLE: Console media-type USB disabled by
 system configuration, media-type reverted to RJ45.
```

この時点では、スタックの USB コンソールは入力を持てません。ログのエントリは、コンソールケーブルが接続されたときを示します。USB コンソールケーブルが switch 2 に接続されると、入力は提供されません。

```
*Mar  1 00:34:27.498: %USB_CONSOLE-6-CONFIG_DISALLOW: Console media-type USB is disallowed
 by system configuration, media-type remains RJ45. (switch-stk-2)
```

次に、前の設定を逆にして、接続されている USB コンソールをただちにアクティブにする例を示します。

```
Device# configure terminal
Device(config)# line console 0
Device(config-line)# no media-type rj45 switch 1
```

例：USB 無活動タイムアウトの設定

次に、無活動タイムアウトを 30 分に設定する例を示します。

```
Device# configure terminal
Device(config)# line console 0
Device(config-line)# usb-inactivity-timeout switch 1 30
```

次に、設定を無効にする例を示します。

```
Device# configure terminal
Device(config)# line console 0
Device(config-line)# no usb-inactivity-timeout switch 1
```

設定された分数の間に USB コンソール ポートで（入力）アクティビティがなかった場合、無活動タイムアウト設定が RJ-45 ポートに適用され、ログにこの発生が示されます。

```
*Mar  1 00:47:25.625: %USB_CONSOLE-6-INACTIVITY_DISABLE: Console media-type USB disabled
 due to inactivity, media-type reverted to RJ45.
```

この時点で、USB コンソール ポートを再度アクティブ化する唯一の方法は、ケーブルを取り外し、再接続することです。

スイッチの USB ケーブルが取り外され、再度接続された場合、次のようなログが表示されます。

```
*Mar  1 00:48:28.640: %USB_CONSOLE-6-MEDIA_USB: Console media-type is USB.
```

インターフェイス特性の設定のその他の関連資料

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 9200 Series Switches)* の「Interface and Hardware Commands」の項を参照してください。

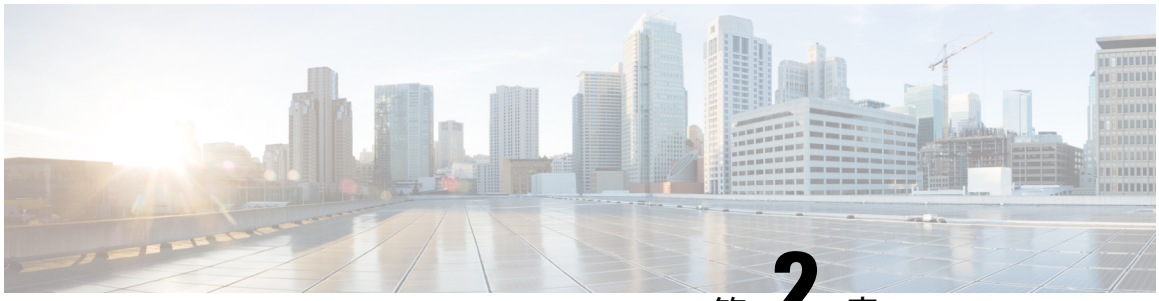
インターフェイス特性の設定の機能履歴

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 16.9.2	インターフェイス特性	インターフェイス特性には、インターフェイスタイプ、接続、設定モード、速度、およびデバイスの物理インターフェイスの設定に関するその他の側面が含まれます。 この機能のサポートは、Cisco Catalyst 9200 シリーズスイッチの 9200L 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。
Cisco IOS XE Gibraltar 16.11.1	マルチギガビットイーサネットインターフェイス	シリーズのすべてのモデルで、100 Mb/s、1 Gb/s、2.5 Gb/s、5 Gb/s、および 10 Gb/s で動作するマルチギガビット イーサネット ポートがサポートされるようになりました。

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 2 章

Auto-MDIX の設定

Auto-MDIX の前提条件

インターフェイスがレイヤ 3 モードの場合に、レイヤ 2 パラメータを設定するには、パラメータを指定せずに **switchport** 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、インターフェイスをレイヤ 2 モードにする必要があります。これにより、インターフェイスがいったんシャットダウンしてから再度有効になり、インターフェイスが接続しているデバイスに関するメッセージが表示されることがあります。レイヤ 3 モードのインターフェイスをレイヤ 2 モードにした場合、影響のあるインターフェイスに関連する以前の設定情報が消失する可能性があり、インターフェイスはデフォルト設定に戻ります。

デフォルトで Automatic Medium-Dependent Interface Crossover（Auto-MDIX）機能が有効に設定されます。

Auto-MDIX の制約事項

受電デバイスがクロスケーブルでデバイスに接続されている場合、そのデバイスは IEEE 802.3af に完全には準拠しておらず、Cisco IP Phone やアクセスポイントなどの準規格の受電デバイスをサポートしていない場合があります。これは、スイッチ ポート上で Automatic Medium-Dependent Interface Crossover（Auto-MIDX）が有効かどうかは関係ありません。

Auto-MDIX の設定について

インターフェイスでの Auto-MDIX

自動メディア依存型インターフェイスクロスオーバー（MDIX）が有効になっているインターフェイスでは、必要なケーブル接続タイプ（ストレートまたはクロス）が自動的に検出され、接続が適切に設定されます。Auto-MDIX機能を使用せずにデバイスを接続する場合、サーバ、ワークステーション、ルータなどのデバイスの接続にはストレートケーブルを使用し、他のデバイスやリピーターの接続にはクロスケーブルを使用する必要があります。Auto-MDIXが有効になっている場合、他のデバイスとの接続にはどちらのケーブルでも使用でき、ケーブルが正しくない場合はインターフェイスが自動的に修正を行います。ケーブル接続の詳細については、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。

（注） Auto-MDIX はデフォルトで有効になっています。

次の表に、Auto-MDIX の設定およびケーブル接続ごとのリンク ステートを示します。

表 *4:* リンク状態と *Auto-MDIX* の設定

ローカル側の Auto-MDIX	リモート側の Auto-MDIX	ケーブル接続が正しい場合	ケーブル接続が正しくない場合
オン	オン	リンク アップ	リンク アップ
オン	オフ	リンク アップ	リンク アップ
オフ	オン	リンク アップ	リンク アップ
消灯	消灯	リンク アップ	リンク ダウン

Auto-MDIX の設定方法

インターフェイスでの Auto-MDIX の設定

デフォルトで Auto MDIX はオンです。ポートで Auto MDIX を無効にするには、インターフェイス コンフィギュレーション モードで **no mdix auto** コマンドを使用します。デフォルトに戻すには、インターフェイス コンフィギュレーション モードで **mdix auto** コマンドを使用します。次に、Auto MDIX を有効にする手順を示します。

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 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-id* 例： `Device(config)# interface gigabitethernet1/0/1`	設定する物理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 4	**mdix auto** 例： `Device(config-if)# mdix auto`	Auto MDIX 機能を有効にします。
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	**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。

Auto-MDIX の設定例

次の例では、ポートの Auto MDIX を有効にする方法を示します。

```
Device# configure terminal
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# mdix auto
Device(config-if)# end
```

Auto-MDIX と動作状態

表 *5 : Auto-MDIX* と動作状態

インターフェイスでの **Auto-MDIX** 設定と動作状態	説明
Auto-MDIX on (operational: on)	Auto-MDIX は有効になっており、フル機能しています。
Auto-MDIX on (operational: off)	このインターフェイスでは Auto-MDIX は有効になっていますが、機能していません。Auto-MDIX 機能を正常に動作させるには、インターフェイス速度を自動ネゴシエーションに設定する必要があります。
Auto-MDIX off	**no mdix auto** コマンドにより、Auto-MDIX が無効になっています。

Auto-MDIX に関するその他の関連資料

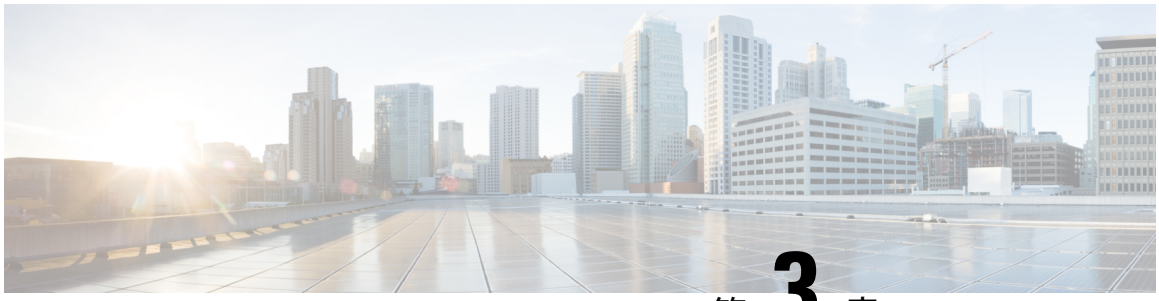
Auto-MDIX の機能履歴

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 16.9.2	インターフェイスでの Auto-MDIX	自動メディア依存型インターフェイスクロスオーバー（Auto-MDIX）対応のインターフェイスは必要なケーブル接続タイプ（ストレートまたはクロス）を自動的に検出し、接続を適切に設定します。

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

イーサネット管理ポートの設定

イーサネット管理ポートの前提条件

PC をイーサネット管理ポートに接続するときに、最初に IP アドレスを割り当てる必要があります。

イーサネット管理ポートについて

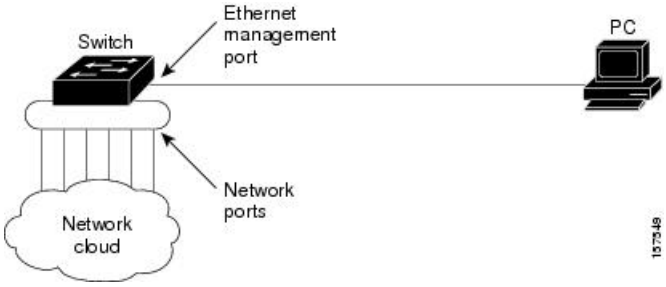
Gi0/0 または *GigabitEthernet0/0* ポートとも呼ばれるイーサネット管理ポートは、PC を接続する VRF（VPN ルーティング/転送）インターフェイスです。ネットワークの管理にデバイスコンソールポートの代わりとしてイーサネット管理ポートを使用できます。

デバイススタックを管理するときに、PC をスタックメンバ上のイーサネット管理ポートに接続します。

デバイスへのイーサネット管理ポートの直接接続

図 *2 : PC* へのデバイスの接続

次の図に、デバイスまたはスタンドアロンデバイス用に PC をイーサネット管理ポートに接続する方法を示します。



ハブを使用したスタックデバイスへのイーサネット管理ポートの接続

スタックデバイスのみが含まれるスタックでは、スタックメンバーのイーサネット管理ポートはすべて、PC が接続されているハブに接続されます。アクティブなスイッチのイーサネット管理ポートからのアクティブなリンクは、ハブを経由して PC とつながっています。アクティブなスイッチに障害が発生し、アクティブなデバイスが新たに選択された場合、アクティブなリンクはその新しいアクティブなデバイス上のイーサネット管理ポートから PC までとなります。

図 *3: PC* へのデバイススタックの接続

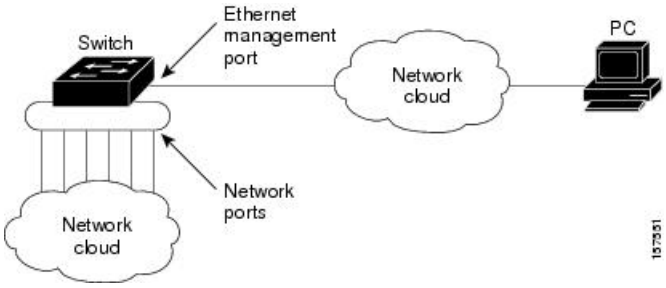
次の図に、PC がハブを使用してデバイススタックに接続する方法を示します。

イーサネット管理ポートおよびルーティング

デフォルトでは、イーサネット管理ポートは有効です。デバイスは、イーサネット管理ポートからネットワークポートへ、およびその逆に、パケットをルーティングできません。イーサネット管理ポートはルーティングをサポートしていませんが、ポート上でルーティングプロトコルを有効にすることが必要となる場合もあります。

図 *4:* ルーティング プロトコルを有効にしたネットワーク例

PC とデバイスが複数ホップ分離れていて、パケットが PC に到達するには複数のレイヤ 3 デバイスを経由する必要がある場合、イーサネット管理ポート上のルーティングプロトコルを有効にします。せ



上記の図では、イーサネット管理ポートとネットワークポートが同じルーティングプロセスに関連付けられている場合、ルートは次のように伝達されます。

- イーサネット管理ポートからのルートは、ネットワーク ポートを通してネットワークに伝播されます。
- ネットワーク ポートからのルートは、イーサネット管理ポートを通してネットワークに伝播されます。

イーサネット管理ポートとネットワーク ポートの間ではルーティングはサポートされていないため、これらのポート間のトラフィックの送受信はできません。このような状況になると、これらのポート間にデータパケットループが発生し、スイッチおよびネットワークの動作が中断されます。このループを防止するには、イーサネット管理ポートとネットワーク ポートの間のルートを回避するためにルート フィルタを設定してください。

サポートされるイーサネット管理ポートの機能

イーサネット管理ポートは次の機能をサポートします。

- Express Setup（デバイススタック内のみ）
- Network Assistant
- パスワード付きの Telnet
- TFTP
- セキュア シェル（SSH）
- Dynamic Host Configuration Protocol（DHCP）ベースの自動設定
- SNMP（ENTITY-MIB および IF-MIB だけ）
- IP ping
- インターフェイス機能
  - 速度：10 Mb/秒、100 Mb/秒、および自動ネゴシエーション
  - デュプレックス モード：全二重、半二重、自動ネゴシエーション
  - ループバック検出
- Cisco Discovery Protocol（CDP）
- DHCP リレー エージェント
- IPv4 および IPv6 アクセス コントロール リスト（ACL）

注意 イーサネット管理ポートの機能を有効にする前に機能がサポートされていることを確認してください。イーサネット管理ポートでサポートされていない機能を設定しようとすると、機能は正しく動作せず、デバイスに障害が発生するおそれがあります。

イーサネット管理ポートの設定方法

イーサネット管理ポートの無効化および有効化

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 gigabitethernet0/0** 例： `Device(config)# interface gigabitethernet0/0`	CLI でイーサネット管理ポートを指定します。
ステップ 3	**shutdown** 例： `Device(config-if)# shutdown`	イーサネット管理ポートを無効にします。
ステップ 4	**no shutdown** 例： `Device(config-if)# no shutdown`	イーサネット管理ポートを有効にします。
ステップ 5	**exit** 例： `Device(config-if)# exit`	インターフェイス コンフィギュレーション モードを終了します。
ステップ 6	**show interfaces gigabitethernet0/0** 例： `Device# show interfaces gigabitethernet0/0`	リンク ステータスを表示します。 PC へのリンク ステータスを調べるには、イーサネット管理ポートの LED をモニタします。リンクがアクティブな場合、LED はグリーン（オン）であり、リンクが停止中の場合は、LED はオフです。POST エラーがある場合は、LED はオレンジです。

次のタスク

イーサネット管理ポートを使用したデバイスの管理または設定に進みます。「ネットワーク管理」の項を参照してください。

イーサネット管理インターフェイスでのIPアドレスの設定例

次に、GigabitEthernet0/0 管理インターフェイスで IP アドレスを設定する例を示します。

```
Device# configure terminal
Device(config)# interface gigabitethernet0/0
Device(config-if)# vrf forwarding Mgmt-vrf
Device(config-if)#ip address 192.168.247.10 255.255.0.0
Device(config-if)# end


Device# show running-config interface Gi0/0
Building configuration...

Current configuration : 118 bytes
!
interface GigabitEthernet0/0
vrf forwarding Mgmt-vrf
ip address 192.168.247.10 255.255.0.0
negotiation auto
end
```

次に、TenGigabitEthernet0/1 管理インターフェイスで IP アドレスを設定する例を示します。

```
Device# configure terminal
Device(config)# interface TenGigabitEthernet0/1
Device(config-if)# vrf forwarding Mgmt-vrf
Device(config-if)#ip address 192.168.247.20 255.255.0.0
Device(config-if)# negotiation auto
Device(config-if)# end


Device#show running-config interface Te0/1
Building configuration...

Current configuration : 118 bytes
!
interface TenGigabitEthernet0/1
 vrf forwarding Mgmt-vrf
 ip address 192.168.247.20 255.255.0.0
 negotiation auto
end
```

イーサネット管理ポートのその他の関連資料

関連資料

関連項目	マニュアル タイトル
ブートローダ設定	このガイドの「システム管理」の項を参照してください。
ブートローダ コマンド	『*Command Reference (Catalyst 9600 Series Switches)*』の「*System Management Commands*」の項を参照

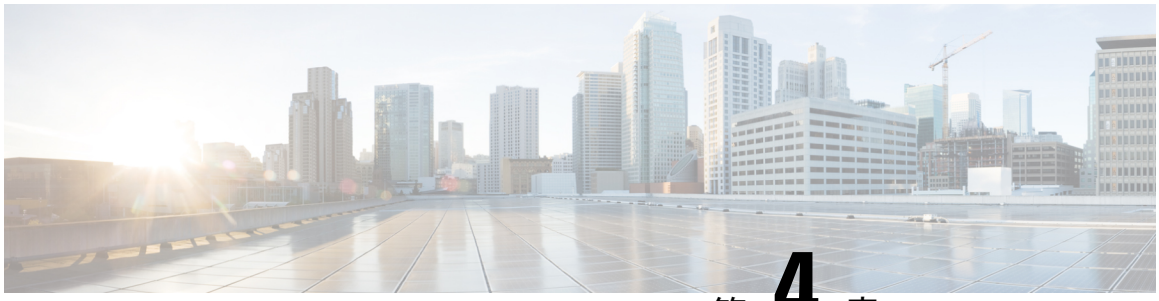
イーサネット管理ポートの機能履歴

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 Fuji 16.9.2	イーサネット管理ポート	イーサネット管理ポートは、PC を接続できる VRF インターフェイスです。ネットワークの管理にデバイスコンソールポートの代わりとしてイーサネット管理ポートを使用できます。

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

ポート ステータスと接続の確認

タイムドメイン反射率計を使用したケーブルステータスの確認

タイムドメイン反射率計（TDR）機能を使用すると、障害発生時にケーブルがOPENかSHORTかを判断できます。

TDR テストの実行

TDR テストを開始するには、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	**test cable-diagnostics tdr** {**interface** { *interface-number* }}	TDR テストを開始します。
ステップ 2	**show cable-diagnostics tdr** { **interface** *interface-number*}	TDR テストのカウンタ情報を表示します。

TDR に関する注意事項

TDR を使用する場合は、次の注意事項が適用されます。

- TDR テストの実行中はポート設定を変更しないでください。
- TDR テストを実行中のポートと Auto-MDIX が有効になっているポートを接続した場合、この TDR 結果は無効となる可能性があります。

- TDR テストを実行中のポートとデバイス上のポートなど 100BASE-T ポートを接続する場合、未使用のペア（4 ～ 5 と 7 ～ 8）はリモートエンドで終端処理されないため、障害として報告されます。
- ケーブルの特性から、正確な結果を入手するには TDR テストを複数回行う必要があります。
- 結果が不正確となる可能性があるため、（近端または遠端のケーブルを取り外すなど）ポート ステータスを変更しないでください。
- TDR は、テスト ケーブルをリモート ポートから外している場合に正しく動作します。それ以外の場合は、正確な結果が得られない可能性があります。
- TDR は 4 本の導線を対象とします。ケーブルの状態によっては、1 組の導線ペアのステータスが OPEN または SHORT と表示され、他のすべてのペアのステータスが faulty と表示される場合があります。この動作は、1 組の導線ペアが OPEN または SHORT であればケーブル不良と宣言する必要があるため、許容範囲です。
- TDR の目的は、不良ケーブルを特定することではなく、ケーブルがどのように不適切な機能をしているかを確認することです。
- TDR でケーブル不良が検出された場合でも、オフライン ケーブル診断ツールを使用して、より詳しく問題を診断する必要があります。

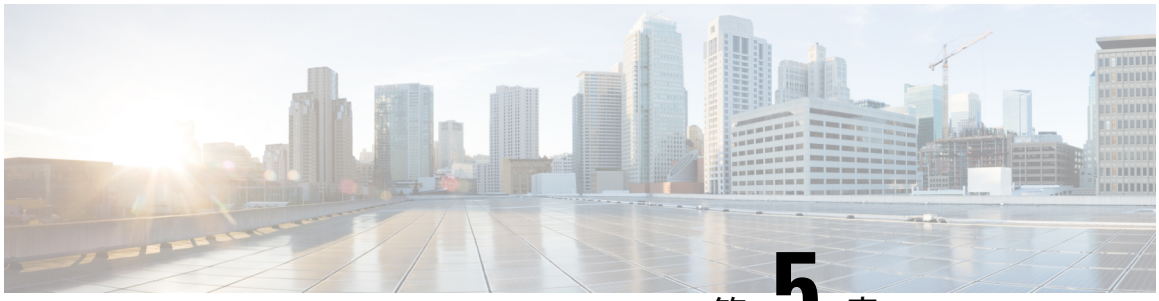
ポートステータスと接続の確認の機能履歴

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	タイムドメイン反射率計（TDR）	TDR を使用すると、障害が発生した場合にケーブルが OPEN か SHORT かを判断できます。

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 5 章

LLDP、LLDP-MED、およびワイヤード ロケーション サービスの設定

LLDP に関する制約事項

- インターフェイスがトンネル ポートに設定されていると、LLDP は自動的に無効になります。
- 最初にインターフェイス上にネットワーク ポリシー プロファイルを設定した場合、インターフェイス上に **switchport voice vlan** コマンドを適用できません。**switchport voice vlan** *vlan-id* がすでに設定されているインターフェイスには、ネットワーク ポリシー プロファイルを適用できます。このように、そのインターフェイスには、音声または音声シグナリング VLAN ネットワーク ポリシー プロファイルが適用されます。
- ネットワーク ポリシープロファイルを持つインターフェイス上では、スタティック セキュア MAC アドレスを設定できません。
- Cisco Discovery Protocol と LLDP が両方とも同じスイッチ内で使用されている場合、Cisco Discovery Protocol が電源ネゴシエーションに使用されているインターフェイスで LLDP を無効にする必要があります。LLDP は、コマンド **no lldp tlv-select power-management** または **no lldp transmit / no lldp receive** を使用してインターフェイス レベルで無効にすることができます。

LLDP、LLDP-MED、およびワイヤード ロケーション サービスについて

LLDP

Cisco Discovery Protocol（CDP）は、すべてのシスコ製デバイス（ルータ、ブリッジ、アクセス サーバ、スイッチ、およびコントローラ）のレイヤ2（データ リンク層）上で動作するデバイス検出プロトコルです。ネットワーク管理アプリケーションは CDP を使用することにより、ネットワーク接続されている他のシスコ デバイスを自動的に検出し、識別できます。

デバイスでは他社製のデバイスをサポートして他のデバイス間の相互運用性を確保するために、IEEE 802.1AB リンク層検出プロトコル（LLDP）をサポートしています。LLDP は、ネットワーク デバイスがネットワーク上の他のデバイスに自分の情報をアドバタイズするために使用するネイバー探索プロトコルです。このプロトコルはデータリンク層で動作するため、異なるネットワーク層プロトコルが稼働する 2 つのシステムで互いの情報を学習できます。

LLDP でサポートされる TLV

LLDP は一連の属性をサポートし、これらを使用してネイバーデバイスを検出します。属性には、Type、Length、および Value の説明が含まれていて、これらを TLV と呼びます。LLDP をサポートするデバイスは、ネイバーとの情報の送受信に TLV を使用できます。このプロトコルは、設定情報、デバイス機能、およびデバイス ID などの詳細情報をアドバタイズできます。

スイッチは、次の基本管理 TLV をサポートします。これらは必須の LLDP TLV です。

- ポート記述 TLV
- システム名 TLV
- システム記述 TLV
- システム機能 TLV
- 管理アドレス TLV

次の IEEE 固有の LLDP TLV もアドバタイズに使用されて LLDP-MED をサポートします。

- ポート VLAN ID TLV（IEEE 802.1 に固有の TLV）
- MAC/PHY コンフィギュレーション/ステータス TLV（IEEE 802.3 に固有の TLV）

LLDP-MED

LLDP for Media Endpoint Devices（LLDP-MED）は LLDP の拡張版で、IP 電話などのエンドポイント デバイスとネットワーク デバイスの間で動作します。特に VoIP アプリケーションをサポートし、検出機能、ネットワーク ポリシー、Power over Ethernet（PoE）、インベントリ管

理、およびロケーション情報に関するTLVを提供します。デフォルトで、すべてのLLDP-MED TLV が有効になります。

LLDP-MED でサポートされる TLV

LLDP-MED では、次の TLV がサポートされます。

- LLDP-MED 機能 TLV

  LLDP-MED エンドポイントは、接続装置がサポートする機能と現在有効になっている機能を識別できます。

- ネットワーク ポリシー TLV

  ネットワーク接続デバイスとエンドポイントはともに、VLAN設定、および関連するレイヤ2とレイヤ3属性をポート上の特定アプリケーションにアドバタイズできます。たとえば、スイッチは使用する VLAN 番号を IP 電話に通知できます。IP 電話は任意のデバイスに接続し、VLAN 番号を取得してから、コール制御との通信を開始できます。

  ネットワーク ポリシー プロファイル TLV を定義することによって、VLAN、サービス クラス（CoS）、Diffserv コード ポイント（DSCP）、およびタギング モードの値を指定して、音声と音声信号のプロファイルを作成できます。その後、これらのプロファイル属性は、スイッチで中央集約的に保守され、IP 電話に伝播されます。

- 電源管理 TLV

  LLDP-MED エンドポイントとネットワーク接続デバイスの間で拡張電源管理を可能にします。デバイスおよび IP 電話は、デバイスの受電方法、電源プライオリティ、デバイスに必要な消費電力などの電源情報を通知することができます。

  LLDP-MED は拡張電源 TLV もサポートして、きめ細かな電力要件、エンドポイント電源プライオリティ、およびエンドポイントとネットワークの接続デバイスの電源ステータスをアドバタイズします。LLDP が有効でポートに電力が供給されているときは、電力 TLV によってエンドポイントデバイスの実際の電力要件が決定するので、それに応じてシステムの電力バジェットを調整することができます。デバイスは要求を処理し、現在の電力バジェットに基づいて電力を許可または拒否します。要求が許可されると、スイッチは電力バジェットを更新します。要求が拒否されると、デバイスはポートへの電力供給をオフにし、Syslogメッセージを生成し、電力バジェットを更新します。LLDP-MEDが無効になっている場合や、エンドポイントが LLDP-MED 電力 TLV をサポートしていない場合は、初期割り当て値が接続終了まで使用されます。

  電力設定を変更するには、**power inline** {**auto** [**max** *max-wattage*] | **never** | **static** [**max** *max-wattage*] } 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。PoE インターフェイスはデフォルトで **auto** モードに設定されています。

- インベントリ管理 TLV

  エンドポイントは、デバイスにエンドポイントの詳細なインベントリ情報を送信できます。インベントリ情報には、ハードウェアリビジョン、ファームウェアバージョン、ソフトウェアバージョン、シリアル番号、メーカー名、モデル名、アセット ID TLV などがあります。

- ロケーション TLV

  デバイスからのロケーション情報をエンドポイントデバイスに提供します。ロケーション TLV はこの情報を送信することができます。

  - 都市ロケーション情報

    都市アドレス情報および郵便番号情報を提供します。都市ロケーション情報の例には、地名、番地、郵便番号などがあります。

  - ELIN ロケーション情報

    発信側のロケーション情報を提供します。ロケーションは、緊急ロケーション識別番号（ELIN）によって決定されます。これは、緊急通報を Public Safety Answering Point（PSAP）にルーティングする電話番号で、PSAP はこれを使用して緊急通報者にコールバックすることができます。

  - 地理的なロケーション情報

    スイッチの緯度、経度、および高度などのスイッチ位置の地理的な詳細を指定します。

  - カスタム ロケーション

    スイッチの位置のカスタマイズされた名前と値を入力します。

ワイヤード ロケーション サービス

デバイスは、接続されているデバイスのロケーション情報およびアタッチメント追跡情報を Cisco Mobility Services Engine（MSE）に送信するのにロケーションサービス機能を使用します。トラッキングされたデバイスは、ワイヤレスエンドポイント、ワイヤードエンドポイント、またはワイヤードデバイスやワイヤードコントローラになります。デバイスは、MSE に Network Mobility Services Protocol（NMSP）のロケーション通知および接続通知を介して、デバイスのリンクアップイベントおよびリンクダウンイベントを通知します。

MSE がデバイスに対して NMSP 接続を開始すると、サーバポートが開きます。MSE がデバイスに接続する場合は、バージョンの互換性を確保する 1 組のメッセージ交換およびサービス交換情報があり、その後にロケーション情報の同期が続きます。接続後、デバイスは定期的にロケーション通知および接続通知を MSE に送信します。インターバル中に検出されたリンクアップ イベントまたはリンク ダウン イベントは、集約されてインターバルの最後に送信されます。

デバイスがリンクアップイベントまたはリンクダウンイベントでデバイスの有無を確認した場合は、スイッチは、MAC アドレス、IP アドレス、およびユーザ名のようなクライアント固有情報を取得します。クライアントが LLDP-MED または CDP に対応している場合は、デバイスは LLDP-MED ロケーション TLV または CDP でシリアル番号および UDI を取得します。

デバイス機能に応じて、デバイスは次のクライアント情報をリンクアップ時に取得します。

- ポート接続で指定されたスロットおよびポート。

- クライアント MAC アドレスで指定された MAC アドレス。
- ポート接続で指定された IP アドレス。
- 802.1X ユーザ名（該当する場合）。
- デバイス カテゴリは、*wired station* として指定されます。
- ステートは *new* として指定されます。
- シリアル番号、UDI。
- モデル番号。
- デバイスによる関連付け検出後の時間（秒）。

デバイス機能に応じて、デバイスは次のクライアント情報をリンクダウン時に取得します。

- 切断されたスロットおよびポート。
- MAC アドレス
- IP アドレス
- 802.1X ユーザ名（該当する場合）。
- デバイス カテゴリは、*wired station* として指定されます。
- ステートは *delete* として指定されます。
- シリアル番号、UDI。
- デバイスによる関連付け解除検出後の時間（秒）。

デバイスがシャットダウンするときに、MSE との NMSP 接続が終了する前に、ステート *delete* および IP アドレスとともに接続情報通知が送信されます。MSE は、この通知をデバイスに関連付けられているすべてのワイヤードクライアントに対する関連付け解除として解釈します。

デバイス上のロケーションアドレスを変更すると、デバイスは、影響を受けるポートを識別する NMSP ロケーション通知メッセージ、および変更されたアドレス情報を送信します。

デフォルトの LLDP 設定

表 *6 :* デフォルトの *LLDP* 設定

機能	デフォルト設定
LLDP グローバル ステート	無効
LLDP ホールドタイム（廃棄までの時間）	120 秒
LLDP タイマー（パケット更新頻度）	30 秒

機能	デフォルト設定
LLDP 再初期化遅延	2 秒
LLDP tlv-select	無効（すべての TLV との送受信）
LLDP インターフェイス ステート	無効
LLDP 受信	無効
LLDP 転送	無効
LLDP med-tlv-select	無効（すべての LLDP-MED TLV への送信）。LLDP がグローバルに有効になると、LLDP-MED-TLV も有効になります。

LLDP、LLDP-MED、およびワイヤード ロケーション サービスの設定方法

LLDP の有効化

手順

	コマンドまたはアクション	目的
ステップ 1	**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 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	**lldp run** 例： `Device(config)# lldp run`	デバイスで LLDP をグローバルに有効にします。

	コマンドまたはアクション	目的
ステップ 4	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet2/0/1`	LLDP を有効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 5	**lldp transmit** 例： `Device(config-if)# lldp transmit`	LLDP パケットを送信するようにインターフェイスを有効にします。
ステップ 6	**lldp receive** 例： `Device(config-if)# lldp receive`	LLDP パケットを受信するようにインターフェイスを有効にします。
ステップ 7	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。
ステップ 8	**show lldp** 例： `Device# show lldp`	設定を確認します。
ステップ 9	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

LLDP 特性の設定

LLDP 更新の頻度、情報を廃棄するまでの保持期間、および初期化遅延時間を設定できます。送受信する LLDP および LLDP-MED TLV も選択できます。

（注） ステップ 3 ～ 6 は任意であり、どの順番で実行してもかまいません。

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。 • 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 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	**lldp holdtime** *seconds* 例： `Device(config)# lldp holdtime 120`	（任意）デバイスから送信された情報を受信側デバイスが廃棄するまで保持する必要がある期間を指定します。 指定できる範囲は 0 ～ 65535 秒です。デフォルトは 120 秒です。
ステップ 4	**lldp reinit** *delay* 例： `Device(config)# lldp reinit 2`	（任意）任意のインターフェイス上で LLDP の初期化の遅延時間（秒）を指定します。 指定できる範囲は 2 ～ 5 秒です。デフォルトは 2 秒です。
ステップ 5	**lldp timer** *rate* 例： `Device(config)# lldp timer 30`	（任意）インターフェイス上で LLDP の更新の遅延時間（秒）を指定します。 指定できる範囲は 5 ～ 65534 秒です。デフォルトは 30 秒です。
ステップ 6	**lldp tlv-select** 例： `Device(config)# tlv-select`	（任意）送受信する LLDP TLV を指定します。
ステップ 7	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet2/0/1`	LLDP を有効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。

	コマンドまたはアクション	目的
ステップ 8	**lldp med-tlv-select** 例： `Device(config-if)# lldp med-tlv-select inventory management`	（任意）送受信する LLDP-MED TLV を指定します。
ステップ 9	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。
ステップ 10	**show lldp** 例： `Device# show lldp`	設定を確認します。
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。

LLDP-MED TLV の設定

デフォルトでは、デバイスはエンドデバイスからLLDP-MEDパケットを受信するまで、LLDP パケットだけを送信します。スイッチは、MED TLV を持つ LLDP も送信します。LLDP-MED エントリが期限切れになった場合は、スイッチは再び LLDP パケットだけを送信します。

lldp 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、インターフェイスが次の表にリストされている TLV を送信しないように設定できます。

表 *7 : LLDP-MED TLV*

LLDP-MED TLV	説明
inventory-management	LLDP-MED インベントリ管理 TLV
location	LLDP-MED ロケーション TLV
network-policy	LLDP-MED ネットワーク ポリシー TLV
power-management	LLDP-MED 電源管理 TLV

インターフェイスで TLV を有効にするには、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 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-id* 例： `Device(config)# interface gigabitethernet2/0/1`	LLDP を有効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 4	**lldp med-tlv-select** 例： `Device(config-if)# lldp med-tlv-select inventory management`	有効にする TLV を指定します。
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	**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。

Network-Policy TLV の設定

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 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	**network-policy profile** *profile number* 例： `Device(config)# network-policy profile 1`	ネットワーク ポリシープロファイル番号を指定し、ネットワーク ポリシー コンフィギュレーション モードを開始します。指定できる範囲は 1 ～ 4294967295 です。
ステップ 4	{**voice** | **voice-signaling**} **vlan** [*vlan-id* { **cos** *cvalue* | **dscp** *dvalue*}] | [[**dot1p** { **cos** *cvalue* | **dscp** *dvalue*}] | **none** | **untagged**] 例： `Device(config-network-policy)# voice vlan 100 cos 4`	ポリシー属性の設定: • **voice**：音声アプリケーション タイプを指定します。 • **voice-signaling**：音声シグナリング アプリケーション タイプを指定します。 • **vlan**：音声トラフィックのネイティブ VLAN を指定します。 • *vlan-id*：（任意）音声トラフィックの VLAN を指定します。指定できる範囲は 1 ～ 4094 です。 • **cos** *cvalue*：（任意）設定された VLAN に対するレイヤ 2 プライオリティ サービス クラス（CoS）を指定します。指定できる範囲は 0 ～ 7 です。デフォルト値は 5 です。 • **dscp** *dvalue*：（任意）設定された VLAN に対する DiffServ コード ポ

	コマンドまたはアクション	目的
		イント（DSCP）値を指定します。指定できる範囲は 0 ～ 63 です。デフォルト値は 46 です。 • **dot1p**：（任意）IEEE 802.1p プライオリティ タギングおよび VLAN 0（ネイティブ VLAN）を使用するように電話を設定します。 • **none**：（任意）音声 VLAN に関して IP Phone に指示しません。IP Phone のキー パッドから入力された設定を使用します。 • **untagged**：（任意）IP Phone を、タグなしの音声トラフィックを送信するよう設定します。これが IP Phone のデフォルト設定になります。
ステップ 5	**exit** 例： `Device(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 6	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet2/0/1`	ネットワーク ポリシープロファイルを設定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 7	**network-policy** *profile number* 例： `Device(config-if)# network-policy 1`	ネットワーク ポリシープロファイル番号を指定します。
ステップ 8	**lldp med-tlv-select network-policy** 例： `Device(config-if)# lldp med-tlv-select network-policy`	ネットワーク ポリシー TLV 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	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 10	**show network-policy profile** 例： `Device# show network-policy profile`	設定を確認します。
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。

ロケーション TLV およびワイヤード ロケーション サービスの設定

エンドポイントのロケーション情報を設定し、その設定をインターフェイスに適用するには、特権 EXEC モー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	**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	**location** { **admin-tag** *string* | **civic-location identifier** {*id* | **host**} | **elin-location** *string* **identifier** *id* | **custom-location identifier** {*id* | **host**} | **geo-location identifier** {*id* | **host**}} 例： `Device(config)# location civic-location identifier 1` `Device(config-civic)# number 3550` `Device(config-civic)# primary-road-name "Cisco Way"`	エンドポイントにロケーション情報を指定します。 • **admin-tag**：管理タグまたはサイト情報を指定します。 • **civic-location**：都市ロケーション情報を指定します。 • **elin-location**：緊急ロケーション情報（ELIN）を指定します。

	コマンドまたはアクション	目的
	Device(config-civic)# **city "San Jose"** Device(config-civic)# **state CA** Device(config-civic)# **building 19** Device(config-civic)# **room C6** Device(config-civic)# **county "Santa Clara"** Device(config-civic)# **country US**	• **custom-location**：カスタム ロケーション情報を指定します。 • **geo-location**：地理空間のロケーション情報を指定します。 • **identifier** *id*：都市、ELIN、カスタム、または地理ロケーションの ID を指定します。 • **host**：ホストの都市、カスタム、または地理ロケーションを指定します。 • *string*：サイト情報またはロケーション情報を英数字形式で指定します。
ステップ 3	**exit** 例： Device(config-civic)# **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 4	**interface** *interface-id* 例：	ロケーション情報を設定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 5	**location** { **additional-location-information** *word* | **civic-location-id** {*id* | **host**} | **elin-location-id** *id* | **custom-location-id** {*id* | **host**} | **geo-location-id** {*id* | **host**} } 例： Device(config-if)# **location elin-location-id 1**	インターフェイスのロケーション情報を入力します。 • **additional-location-information**：ロケーションまたは場所に関する追加情報を指定します。 • **civic-location-id**：インターフェイスにグローバル都市ロケーション情報を指定します。 • **elin-location-id**：インターフェイスに緊急ロケーション情報を指定します。 • **custom-location-id**：インターフェイスにカスタム ロケーション情報を指定します。

	コマンドまたはアクション	目的
		• **geo-location-id**：インターフェイスに地理空間のロケーション情報を指定します。 • **host**：ホストのロケーション ID を指定します。 • *word*：追加のロケーション情報を指定する語またはフレーズを指定します。 • *id*：都市、ELIN、カスタム、または地理ロケーションの ID を指定します。指定できる ID 範囲は 1 ～ 4095 です。
ステップ 6	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。
ステップ 7	次のいずれかを使用します。 • **show location admin-tag** *string* • **show location civic-location identifier** *id* • **show location elin-location identifier** *id* 例： `Device# show location admin-tag` または `Device# show location civic-location identifier` または `Device# show location elin-location identifier`	設定を確認します。
ステップ 8	**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`	

デバイスでのワイヤード ロケーション サービスの有効化

始める前に

ワイヤードロケーションが機能するためには、まず、**ip device tracking** グローバル コンフィギュレーション コマンドを入力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。 • 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 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	**nmsp notification interval** {**attachment** | **location**} *interval-seconds* 例： `Device(config)# nmsp notification interval location 10`	NMSP 通知間隔を指定します。 **attachment**：接続通知間隔を指定します。 **location**：ロケーション通知間隔を指定します。 *interval-seconds*：デバイスから MSE にロケーション更新または接続更新が送信されるまでの期間（秒）。指定できる範囲は 1 ～ 30 です。デフォルト値は 30 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 network-policy profile** 例： `Device# show network-policy profile`	設定を確認します。
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。

LLDP、LLDP-MED、およびワイヤード ロケーション サービスの設定例

Network-Policy TLV の設定：例

次に、CoS を持つ音声アプリケーションの VLAN 100 を設定して、インターフェイス上のネットワーク ポリシー プロファイルおよびネットワーク ポリシー TLV を有効にする例を示します。

```
Device# configure terminal
Device(config)# network-policy 1
Device(config-network-policy)# voice vlan 100 cos 4
Device(config-network-policy)# exit
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# network-policy profile 1
Device(config-if)# lldp med-tlv-select network-policy
```

次の例では、プライオリティ タギングを持つネイティブ VLAN 用の音声アプリケーション タイプを設定する方法を示します。

```
Device-config-network-policy)# voice vlan dot1p cos 4

Device-config-network-policy)# voice vlan dot1p dscp 34
```

LLDP、LLDP-MED、ワイヤード ロケーション サービスのモニタリングとメンテナンス

以下は、LLDP、LLDP-MED、ワイヤードロケーションサービスのモニタリングとメンテナンスのコマンドです。

コマンド	説明
clear lldp counters	トラフィック カウンタを0にリセットします。
clear lldp table	LLDP ネイバー情報テーブルを削除します。
clear nmsp statistics	NMSP 統計カウンタをクリアします。
show lldp	送信頻度、送信するパケットのホールドタイム、LLDP 初期化の遅延時間のような、インターフェイス上のグローバル情報を表示します。
show lldp entry *entry-name*	特定のネイバーに関する情報を表示します。 アスタリスク（*）を入力すると、すべてのネイバーの表示、またはネイバーの名前の入力が可能です。
show lldp interface [*interface-id*]	LLDPが有効になっているインターフェイスに関する情報を表示します。 表示対象を特定のインターフェイスに限定できます。
show lldp neighbors [*interface-id*] [**detail**]	デバイス タイプ、インターフェイスのタイプや番号、ホールドタイム設定、機能、ポート ID など、ネイバーに関する情報を表示します。 特定のインターフェイスに関するネイバー情報だけを表示したり、詳細表示にするため表示内容を拡張したりできます。
show lldp traffic	送受信パケットの数、廃棄したパケットの数、認識できない TLV の数など、LLDP カウンタを表示します。
show location admin-tag *string*	指定した管理タグまたはサイトのロケーション情報を表示します。

コマンド	説明
show location civic-location identifier *id*	特定のグローバル都市ロケーションのロケーション情報を表示します。
show location elin-location identifier *id*	緊急ロケーションのロケーション情報を表示します。
show network-policy profile	設定されたネットワークポリシー プロファイルを表示します。
show nmsp	NMSP 情報を表示します。

LLDP、LLDP-MED、およびワイヤード ロケーション サービスの追加情報

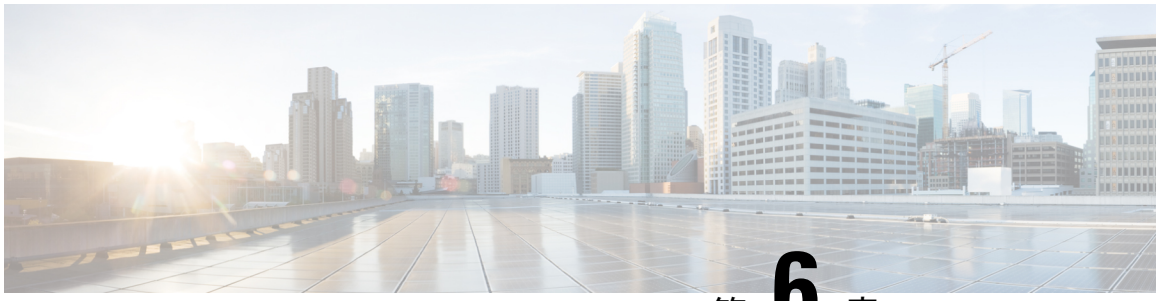
LLDP、LLDP-MED、およびワイヤード ロケーション サービスの機能履歴

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 Fuji 16.9.2	Link Layer Discovery Protocol（LLDP）、LLDP-MED、ワイヤード ロケーション サービス	LLDP は、ネットワーク デバイスがネットワーク上の他のデバイスに自分の情報をアドバタイズするために使用するネイバー探索プロトコルです。このプロトコルはデータリンク層で動作するため、異なるネットワーク層プロトコルが稼働する 2 つのシステムで互いの情報を学習できます。 LLDP-MED はエンドポイントとネットワークデバイス間で動作します。 ワイヤード ロケーション サービスでは、接続されているデバイスの追跡情報を Cisco Mobility Services Engine（MSE）に送信できます。

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 6 章

システム MTU の設定

MTU について

イーサネットフレームで受信し、すべてのデバイスインターフェイスで送信されるペイロードのデフォルトの最大伝送ユニット（MTU）サイズは 1500 バイトです。

システム MTU 値の適用

IP または IPv6 MTU 値の上限は、スイッチまたはスイッチ スタックの設定に基づき、現在適用されているシステム MTU 値を参照します。MTU サイズの設定に関する詳細については、このリリースのコマンドリファレンスで **system mtu** グローバル コンフィギュレーション コマンドを参照してください。

MTU の設定方法

システム MTU の設定

スイッチド パケットの MTU サイズを変更するには、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 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	**system mtu** *bytes* 例： `Device(config)# system mtu 1900`	（任意）すべてのインターフェイスの MTU サイズを変更します。
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	**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。
ステップ 6	**show system mtu** 例： `Device# show system mtu`	設定を確認します。

プロトコル固有 MTU の設定

ルーテッドインターフェイスのシステムMTU値を上書きするには、各ルーテッドインターフェイスでプロトコル固有の MTU を設定します。ルーテッドポートの MTU サイズを変更するには、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。

	コマンドまたはアクション	目的
ステップ 2	**interface** *interface* 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 モードを開始します。
ステップ 3	**ip mtu** *bytes* 例： `Device(config-if)# ip mtu 68`	IPv4 MTU サイズを変更します。
ステップ 4	**ipv6 mtu** *bytes* 例： `Device(config-if)# ipv6 mtu 1280`	（任意）IPv6 MTU サイズを設定します。
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	**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。
ステップ 7	**show system mtu** 例： `Device# show system mtu`	設定を確認します。

システム MTU の設定例

例：プロトコル固有 MTU の設定

```
Device# configure terminal
Device(config)# interface gigabitethernet 0/1
Device(config-if)# ip mtu 900
Device(config-if)# ipv6 mtu 1286
Device(config-if)# end
```

例：システム MTU の設定

```
Device# configure terminal
Device(config)# system mtu 1600
```

```
Device(config)# exit
```

システム MTU 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 9200 Series Switches)* の「*Interface and Hardware Commands*」の項を参照してください。

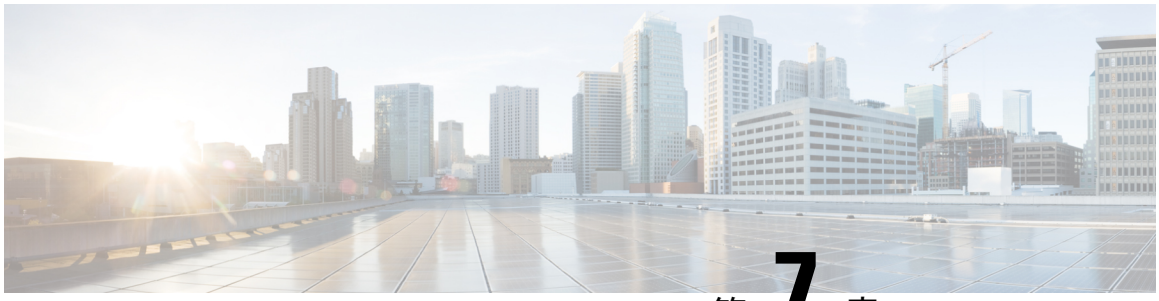
システム MTU の機能履歴

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 16.9.2	システム MTU	システム MTU は、スイッチのすべてのインターフェイスで送信されるフレームの最大伝送ユニットサイズを定義します。

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 7 章

ポート単位の MTU の設定

ポート単位の MTU の制約事項

- ポート単位の MTU は、管理ポートでは設定できません。
- ポート単位の MTU は、SVL リンクでは設定できません。
- ポートチャネルのメンバーはポート単位の MTU を使用して設定できません。ポートチャネルの MTU 設定から MTU を取得します。
- ポート単位の MTU は、サブインターフェイスとポートチャネルサブインターフェイスではサポートされていません。

ポート単位の MTU について

system mtu コマンドを使用して、デバイス上のすべてのインターフェイスの MTU サイズを同時に設定できます。すべてのインターフェイスで送受信されるフレームのデフォルト最大伝送単位（MTU）サイズは、1500 バイトです。**system mtu** コマンドはグローバルコマンドであり、MTU をポートレベルで設定することはできません。Cisco IOS XE 17.1.1 以降では、ポート単位の MTU を設定できます。ポート単位の MTU はポートレベルとポートチャネルレベルの MTU 設定をサポートします。ポート単位の MTU を使用すると、異なるインターフェイスと異なるポート チャネル インターフェイスに異なる MTU 値を設定できます。

ポート単位の MTU は 1500 ー 9198 バイトの範囲で設定できます。

ポート単位の MTU の設定

ポートにポート単位の MTU 値が設定されると、そのポートのプロトコル固有の MTU もポート単位の MTU 値に変更されます。ポート上でポート単位の MTU が設定されている場合でも、インターフェイス上でプロトコル固有の MTU を 256 からポート単位の MTU 値の範囲で設定できます。

ポート単位の MTU が無効になっている場合、ポートの MTU はシステムの MTU 値に戻ります。

show interface mtu コマンドを使用して、インターフェイスのポート単位の MTU 設定を表示できます。

インターフェイスでポート単位の MTU 設定が変更された場合は、次のような動作が予期されます。

- ポートチャネルが PAgP モードか LACP モードの場合、インターフェイスがフラップします。
- ポートチャネルが **on** モードの場合、インターフェイスはフラップしません。
- インターフェイスがポートチャネルでない場合、インターフェイスはフラップしません。

インターフェイス コンフィギュレーション モードで **mtu***bytes* コマンドの **no** 形式を使用して、ポート単位の MTU を無効にできます。

ポート単位の MTU の設定

インターフェイスの特定のポートのスイッチドパケットの MTU サイズを変更するには、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 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***typeswitch-number/slot-number/port-number* 例： `Device(config)# int FortyGigabitEthernet2/5/0/20`	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。

	コマンドまたはアクション	目的
ステップ 4	**mtu***bytes* 例： `Device(config-if)# mtu 6666`	インターフェイスの特定のポートの MTU サイズを設定します。
ステップ 5	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。

例：ポート単位の MTU の設定

次に、インターフェイスでポート単位の MTU を設定する例を示します。

```
Device# configure terminal
Device(config)# interface FortyGigabitEthernet2/5/0/20
Device(config-if)# mtu 6666
Device(config-if)# end
```

例：ポート単位の MTU の確認

次に、**show interface mtu** コマンドを使用してインターフェイスのポート単位の MTU を確認する例を示します。

```
Device# show interface mtu
Port           Name               MTU
Fo2/5/0/19                        1500
Fo2/5/0/20                        6666
Fo2/5/0/21      ixia_7_21         1500
```

例：ポート単位の MTU の無効化

次に、インターフェイスでポート単位の MTU を無効にする例を示します。

```
Device# configure terminal
Device(config)# interface FortyGigabitEthernet2/5/0/20
Device(config-if)# no mtu
Device(config-if)# end
```

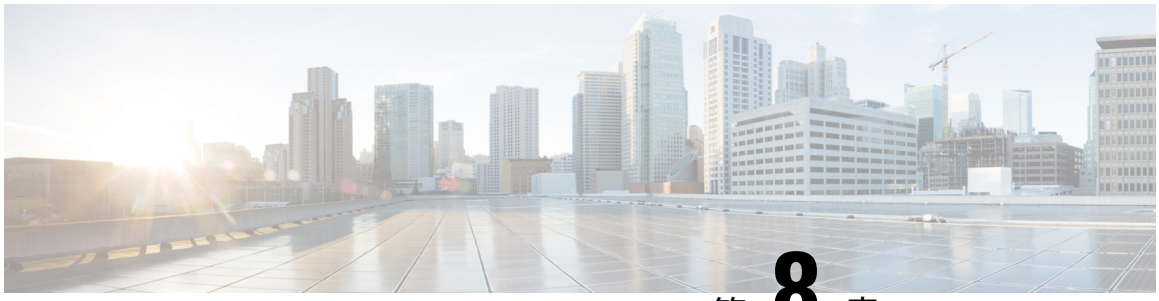
ポート単位の MTU の機能履歴

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 17.1.1	ポート単位の MTU	ポート単位のMTUは、特定のポートまたはポートチャネルで送受信されるフレームの最大伝送ユニットサイズを定義します。 。

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 8 章

内部電源装置の設定

内部電源装置に関する情報

電源装置に関する情報については、デバイスの設置ガイドを参照してください。

内部電源装置の設定方法

内部電源装置の設定

power supply EXEC コマンドを使用すると、デバイスの内部電源装置の設定および管理ができます。コマンドは、 **no power supply** EXEC コマンドをサポートしていません。

ユーザ EXEC モードで開始し、次の手順に従います。

手順

	コマンドまたはアクション	目的
ステップ 1	**power supply** *switch_number* **slot**{**A** | **B**} { **off** | **on** } 例： `Device# power supply 1 slot A on`	次のいずれかのキーワードを使用して、指定した電源装置を **off** または **on** に設定します。 • **A** ：スロット A の電源を選択します。

	コマンドまたはアクション	目的
		• **B** ：スロット B の電源装置を選択します。 （注） 電源装置のスロット B は、デバイスの外側エッジに近いほうです。 • **off** ：電源装置をオフに設定します。 • **on**：電源装置をオンに設定します。 デフォルトでは、デバイスの電源装置は **on** です。
ステップ 2	**show environment power** 例： `Device# show environment power`	設定を確認します。

内部電源装置のモニタ

表 *8 :* 電源装置の *show* コマンド

コマンド	目的
show environment power [**all** | **switch** *switch_number*]	（任意）スタック内の各デバイスまたは指定したデバイスの内部電源装置のステータスを表示します。指定できる範囲は、スタック内のデバイスメンバ番号に従って 1 ～ 8 です。 デバイスキーワードは、スタック対応デバイス上でだけ使用できます。

内部電源装置の設定例

次に、スロット A の電源装置をオフに設定する例を示します。

```
Device# power supply 1 slot A off
Disabling Power supply A may result in a power loss to PoE devices and/or switches
Device#
Jun 10 04:52:54.389: %PLATFORM_ENV-6-FRU_PS_OIR: FRU Power Supply 1 powered off
Jun 10 04:52:56.717: %PLATFORM_ENV-1-FAN_NOT_PRESENT: Fan is not present
Device#
```

次に、スロット A の電源装置をオンに設定する例を示します。

```
Device# power supply 1 slot A on
Jun 10 04:54:39.600: %PLATFORM_ENV-6-FRU_PS_OIR: FRU Power Supply 1 powered on
```

次に、**show env power** コマンドの出力例を示します。

表 *9 : show env power* ステータスの説明

フィールド	説明
OK	電源装置が存在し、電力が良好です。
Not Present	電源装置が未搭載です。
No Input Power	電源装置は存在しますが、入力電力が供給されていません。
Disabled	電源装置が存在し、入力電力は供給されていますが、電源装置が CLI によってオフになっています。
応答なし	電源装置が認識されていないか、障害が発生しています。
Failure-Fan	電源装置のファンに障害が発生しています。

内部電源装置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 9200 Series Switches)*
電源装置に関する情報。	*Cisco Catalyst 9200* シリーズ スイッチ ハードウェア設置ガイド
PoE ポートプライオリティおよび負荷制限についての情報。	『*Interface and Hardware Components Configuration Guide (Catalyst 9200 Series Switches)*』の「*Configuring Interface Characteristics*」の章。

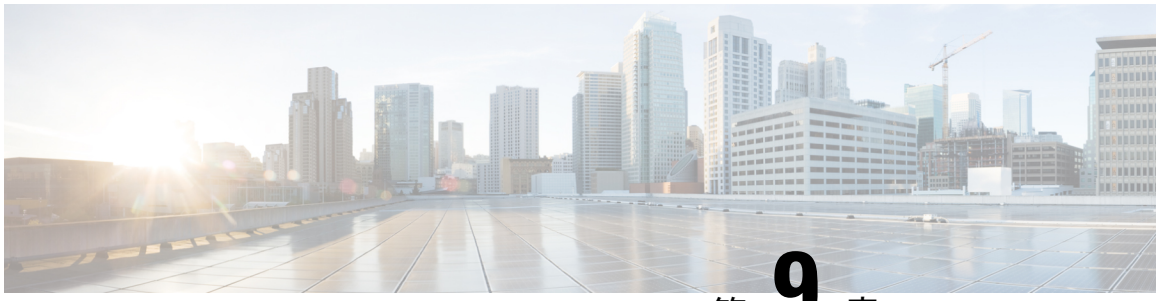
内部電源装置の機能履歴

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 16.9.2	内部電源装置	スイッチは、AC、DC、またはその両方の電源モジュールで動作します。電源装置の詳細については、『*Hardware Installation Guide*』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 には、http://www.cisco.com/go/cfn からアクセスします。



第 9 章

Cisco Expandable Power System 2200 の設定

このモジュールの構成は次のとおりです。

Expandable Power System 2200 の設定に関する制約事項

- スイッチ電源装置をバックアップするために Expandable Power System（XPS）電源装置を RPS モードで使用する場合、XPS の最小ワット数の電源装置は、RPS モードの XPS ポートに接続されているスイッチで最大ワット数の電源装置よりも、ワット数が大きい必要があります。
- RPS モードで各 XPS 電源装置がバックアップできるスイッチ電源装置は、そのサイズにかかわらず、1 台だけです。
- 電源スタックから（スイッチまたは XPS の）電源装置を取り外す場合は、取り外すことによって使用可能な電力が使い尽くされて、負荷制限が発生しないように注意する必要があります。

XPS 2200 の設定について

ここでは、XPS 2200 とその電源モードの概要について説明します。

Cisco eXpandable Power System（XPS）2200 の概要

Cisco eXpandable Power System（XPS）2200 は、独立型電源システムで、Catalyst スイッチに接続できます。XPS 2200 は、接続されている装置で電源装置の故障が発生した場合、その装置

にバックアップ電力を供給できます。また、Catalyst電源スタックでは、電源スタックバジェットに追加の電力を供給できます。XPS 2200の電源ポートと内部電源装置は、冗長電源（RPS）モードまたはスタック電源（SP）モードで動作できます。

スタック電源モードは、電源スタックに属するスタック対応スイッチでのみ使用されます。XPS が含まれていない場合、電源スタックはリング トポロジで動作し、最大 4 台のスイッチで構成できます。2 つのスタックをマージする場合は、スイッチの合計数が 4 台を超えないようにしてください。XPS を電源スタックに追加すると、スタック内で最大 9 台のスイッチと XPS を接続し、スタック電源のリング トポロジ動作と同じような電力バジェットを電源スタックのメンバに提供できます。

SP ポートを経由して XPS に接続されたすべての Catalyst スイッチは同じ電源スタックに属し、XPS とスイッチから供給されるすべての電力はスタック内のすべてのスイッチで共有されます。電源共有がデフォルトのモードですが、XPS は、リング トポロジでサポートされているのと同じスタック電源モード（厳密または厳密でない電源共有モードと冗長モード）をサポートします。

電源装置が 2 台ある場合、1 台を RPS モードにし、もう 1 台を SP モードにするという混在モードで動作させることができます。ポートと電源装置は、XPS 2200 の使用目的に合わせて設定できます。

XPS 2200 には、RPS ロールまたは自動スタック電源（Auto-SP）ロール（デフォルト）で動作できる 9 個の電源ポートがあります。動作モードは、ポートに接続するスイッチの種類によって決まります。CLI を使用して、スタック可能なスイッチに適用するモードを強制的に RPS にすることもできます。

XPS は電源ポートに接続されている任意のスイッチで設定します。任意の XPS ポートを使用して設定でき、XPS に接続されている任意のスイッチから任意のポートを設定できます。複数のスイッチで XPS コンフィギュレーション コマンドを入力した場合、適用された最後の設定が有効になります。

すべての XPS 設定はスイッチで実行できますが、XPS 2200 では専用のソフトウェアが実行されています。このソフトウェアは、XPS サービス ポートを使用してアップグレードできます。

XPS 2200 電源モード

XPS には 2 台の電源装置があり、それぞれ RPS モードまたは SP モードで動作できます。

SP モードでは、XPS のすべての SP ポートは同じ電源スタックに属します。電源スタックに XPS を入れると、スタックのトポロジはスター トポロジになり、最大 9 台のメンバ スイッチと XPS 2200 で構成されます。SP モードの 1 台または 2 台の XPS 電源装置は、電力バジェットの計算で考慮されます。両方の XPS 電源装置が RPS モードの場合、電源スタックは、SP モードの XPS ポートに接続されているスイッチだけで構成され、電力バジェットはそれらのスイッチの電源装置によって決まります。

電源装置のロールに不整合がある場合、たとえば、1 つの XSP ポートが RPS に設定されていて、電源装置が両方とも SP モードの場合、XPS はこの不整合を検出してエラー メッセージを送信します。

RPS モード

両方の XPS 電源装置を RPS モードにすると、XPS は、ワット数が等しいまたは小さいスイッチの電源装置について、2 台の電源装置の故障をバックアップできます。XPS で最小ワット数の電源装置は、RPS モードの XPS ポートに接続されているスイッチで最大ワット数の電源装置よりも、ワット数が大きい必要があります。

1 台の電源装置だけが RPS モードの場合、故障した電源装置のワット数がかなり小さい場合でも XPS がバックアップできるのは 1 台の電源装置だけです。たとえば、XPS 1100 W の電源装置が RPS モードで、2 台の 350 W のスイッチ電源装置が故障した場合、XPS がバックアップできるのは、いずれか一方のスイッチ電源装置だけです。

RPS モードの 1 台の XPS 電源装置がスイッチ電源装置をバックアップしていて、別のスイッチ電源装置が故障した場合、XPS によるバックアップは受けられないというメッセージが表示されます。故障した電源装置が復旧すると、XPS は他の電源装置をバックアップできるようになります。

1 台のスイッチに取り付けられている 2 台の故障した電源装置を XPS がバックアップしている場合（XPS 電源装置は両方とも RPS モード）、故障した電源装置が両方とも修理されるか交換されるまで、XPS は他のスイッチの電源装置をバックアップできません。

1 台の電源装置が RPS モード、もう 1 台が SP モードの混在モードで、1 台のスイッチに取り付けられている 2 台の電源装置が故障した場合、XPS はいずれか一方の電源装置しかバックアップできないので、XPS は両方の電源装置への電力供給を拒否します。このため、スイッチはシャットダウンします。これは混在電源モードでのみ発生します。

スイッチは RPS に設定されているポートに接続されているが、電源装置が両方とも RPS でない場合、RPS ポート設定は拒否され、XPS はスイッチを電源スタックに追加しようとします。スイッチが SP モードで動作できない（スタック可能なスイッチでない）場合、ポートはディセーブルになります。

RPS モードのポートには、プライオリティを設定できます。デフォルトのプライオリティは、XPS ポート番号に基づき、ポート 1 が最もプライオリティが高いポートです。プライオリティの高いポートには、プライオリティの低いポートよりも優先的にバックアップ電力が供給されます。プライオリティの低いポートに接続されているスイッチをバックアップしているときにプライオリティの高いポートに接続されているスイッチで電源装置の故障が発生した場合、XPS は、プライオリティの高いポートに電力を供給するためにプライオリティの低いポートへの電力を削減します。

スタック電源モード

スタック電源モードは、電源スタックに属する Catalyst スイッチでのみ使用します。XPS が含まれていない場合、電源スタックはリング トポロジで動作し、最大 4 台のスイッチで構成できます。XPS を電源スタックに追加すると、スタック内で最大 9 台のスイッチと XPS を接続し、スタック電源のリング トポロジ動作と同じような電力バジェットを電源スタックのメンバに提供できます。

SP ポートを経由して XPS に接続されたすべての Catalyst スイッチは同じ電源スタックに属し、XPS とスイッチから供給されるすべての電力はスタック内のすべてのスイッチで共有されます。電源共有がデフォルトのモードですが、XPS は、リング トポロジでサポートされている

のと同じスタック電源モード（厳密または厳密でない電源共有モードと冗長モード）をサポートします。

XPS はネイバー探索を使用して電源スタックを作成します。XPS は未設定ポートで Catalyst スイッチを検出すると、そのポートを SP ポートとしてマークするので、そのスイッチは電源スタックに追加されます。XPS はスイッチに通知し、電力バジェット配分プロセスを開始し、電源スタックに属するスイッチの要件、プライオリティ、現在の電力割り当て、およびスタック集約電源能力に基づいて各スイッチにバジェットを割り当てます。

XPS は電力バジェットを各スイッチに送信します。各スイッチに必要な最大電力を供給するために使用できる入力電力が足りない場合、電力はプライオリティに基づいて分配されます。最初にプライオリティの最も高いスイッチに必要な電力が分配され、その後にすでに電力が割り当てられているすべての受電デバイスにプライオリティ順に電力が分配されます。残りの電力はスタック全体で均等に分配されます。

RPS ポートのプライオリティ（1 ～ 9）は、スタック電源のプライオリティに影響しません。スタック電源に参加している各スイッチには、独自のシステム プライオリティ、およびそのポートに接続される装置用の高および低プライオリティがあります。これらのプライオリティは、リング トポロジと同様にスタック電源で使用されます。システム、高プライオリティのポート、および低プライオリティのポートにスタック電源のプライオリティを設定するには、スイッチスタック電源コンフィギュレーション モードで **power-priority switch**、**power-priority high**、および **power-priority low** コマンドを使用します。システムまたは一連の受電デバイスがデフォルトのプライオリティを使用している場合、XPS は、自動的にプライオリティ（1 ～ 27）を割り当てます。この際、MAC アドレスの小さいほうに高いプライオリティを割り当てます。

電源スタック モードは、電源共有、厳密な電源共有、冗長、厳密な冗長の 4 つです。電源スタックモードを設定するには、電源スタック コンフィギュレーション モードで **mode** {**power-sharing** | **redundant**} [**strict**] コマンドを使用します。**power-sharing** または **redundant** 設定は、スタックの電力バジェットに影響し、**strict** を指定するかどうかは、バジェットの減少によって負荷制限が発生しないときの PoE アプリケーションの動作に影響します。

- （厳密または厳密でない）電源共有モードの場合、スタックの電力バジェットは、スタック内のすべての電源装置の出力容量を累積した値から 30 W の予約電力を引いた値です。これはデフォルトです。
- （厳密または厳密でない）冗長モードの場合、スタックの電力バジェットは、電源スタックで最大の電源装置の出力容量を引いた後で使用できる合計電力から 30 W を引いた値です。冗長モードでは、1 台の電源装置が故障した場合にスイッチまたは受電デバイスで停電または負荷制限が発生しないことが保証されます。ただし、複数の電源装置が故障した場合、負荷制限が発生する可能性があります。
- 厳密なモードで、入力電力の損失が原因で電力バジェットの減少が発生し、ハードウェアの負荷制限は発生しなかった場合、電力の割り当て量が使用可能な PoE 電力量を下回るか等しくなるまで、XPS は、プライオリティの低いほうから順に受電デバイスへの電力供給を自動的に拒否し始めます。
- 厳密でないモードでは、電力の減少が発生した場合、電力の割り当て量をバジェット内に収めることが許可されます。

たとえば、PoE バジェットの合計（使用可能な電力）が 400 W のシステムは、バジェットから 390 W（割り当て電力）を受電デバイスに割り当てることができます。装置に割り当てる電力は、その装置に必要な最大電力量です。一連の受電デバイスが実際に消費する電力（消費電力）は通常、割り当て電力と等しくなりません。この例では、実際の電力は約 200 W である可能性があります。スタック内での電力損失によって使用可能な電力が 210 W に減った場合、この電力量は受電デバイスが消費する電力を維持するのに十分ですが、最悪の場合の割り当て電力を下回っています。システムはバジェット内に収まります。厳密なモードでは、スタックは、割り当て電力が 210 W 以下になるまで、すぐに受電デバイスへの電力供給を拒否します。厳密でないモードでは、何も動作は行われず、状態を維持できます。厳密でないモードで実際の消費電力が 210 W を上回った場合、これによって負荷制限が発生し、プライオリティ レベルの最も低いすべての受電デバイスまたはスイッチへの電力が失われる可能性があります。

混在モード

XPS 2200 は混在モードでも動作できます。このモードでは、スイッチと接続するポートは RPS と SP の場合があります。この設定では、少なくとも 1 台の電源装置を RPS 電源装置にする必要があります。XPS の電源装置がバックアップできるスイッチ電源装置は、1 台だけです。また、その XPS 電源装置は、RPS モードの XPS ポートに接続されているスイッチで最大ワット数の電源装置よりも、ワット数が大きい必要があります。

SP ポートに接続されたスイッチは、1 つの電源スタックに属します。SP スイッチに十分な大きさの電力バジェットがある場合、XPS に SP 電源装置は必要ありません。XPS 電源装置を設定すると、その電力は電源スタックで共有する電源プールに追加されます。

XPS 2200 システムのデフォルト

XPS 電源装置 A（PS1）のデフォルトは RPS モードです。電源装置 B（PS2）のデフォルトは SP モードです。

すべてのポートと電源装置のデフォルト モードはイネーブルです。

RPS に設定されているポートでは、デフォルトのプライオリティはポート番号と同じです。

Cisco Expandable Power System 2200 の設定方法

XPS は、XPS ポートに接続されている任意のスイッチで設定できます。複数のスイッチで XPS コンフィギュレーション コマンドを入力した場合、適用された最後の設定が有効になります。スイッチ コンフィギュレーション ファイルに保存されるのは、スイッチとポートの名前だけです。

システム名の設定

手順

	コマンドまたはアクション	目的
ステップ 1	**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**	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	**power xps** *switch-number* **name** {*name* | **serialnumber**}	（注） *switch-number* は、Catalyst スイッチにのみ表示され、1～8 の値でデータ スタック内のスイッチ番号を表します。 スタック構成のシステムでは、入力するスイッチ番号に、アクティブ スイッチのスイッチ番号を指定する必要があります。 XPS 2200 システムの名前を設定します。 • *name*：XPS 2200 システムの名前を入力します。名前には最大 20 文字が使用できます。 • **serialnumber**：XPS 2200 のシリアル番号をシステム名として使用します。
ステップ 4	**power xps** *switch-number* **port** {*name* | **hostname** | **serialnumber**}	（注） *switch-number* は、Catalyst スイッチにのみ表示され、1～8 の値でデータスタック内のデバイス番号を表します。 デバイスに接続されている XPS 2200 ポートの名前を設定します。 • *name*：XPS 2200 ポートの名前を入力します。 • **serialnumber**：ポートに接続されているデバイスのシリアル番号を使用します。

	コマンドまたはアクション	目的
		• **hostname**：ポートに接続されているデバイスのホスト名を使用します。
ステップ 5	**end**	特権 EXEC モードに戻ります。
ステップ 6	**show env xps system**	設定したシステムとポートの名前を確認します。
ステップ 7	**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。

XPS ポートの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**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	**power xps** *switch-number* **port** {*number* | **connected**} **mode** {**disable** | **enable**}	（注） *switch-number* は、Catalyst スイッチにのみ表示され、1 ～ 9 の値でデータ スタック内のスイッチ番号を表します。 ポートをイネーブルまたはディセーブルに設定します。 • *number*：XPS 2200 ポート番号を入力します。指定できる範囲は 1 ～ 9 です。 • **connected**：スイッチが接続されているポート番号がわからない場合は、このキーワードを入力します。 • **mode disable**：XPS ポートをディセーブル（シャットダウン）にします。

	コマンドまたはアクション	目的
		（注） XPS ポートをディセーブルにすることは、ケーブルを取り外すことに似ているので、**show** コマンドの出力では同じに見えます。物理的なケーブルが接続されている場合、**enable** キーワードを使用してポートをイネーブルにできます。 • **mode enable**：XPS ポートをイネーブルにします。これはデフォルトです。
ステップ 3	**power xps** *switch-number* **port** {*number* | **connected**} **role** {**auto** | **rps**}	（注） *switch-number* は、1 ～ 9 の値でデータ スタック内のスイッチ番号を表します。 XPS ポートの役割を設定します。 • **role auto**：ポートのモードは、ポートに接続されているスイッチによって決まります。これはデフォルトです。 • **role RPS**：XPS は、スイッチ電源装置が故障した場合にバックアップとして機能します。この設定では、少なくとも 1 台の RPS 電源装置を RPS モードにする必要があります。
ステップ 4	**power xps** *switch-number* **port** {*number* | **connected**} **priority** *port-priority*	（注） *switch-number* は、1 ～ 9 の値でデータ スタック内のスイッチ番号を表します。 ポートの RPS プライオリティを設定します。複数の電源装置が故障した場合、プライオリティの高いポートはプライオリティの低いポートよりも優先されます。このコマンドは、ポートのモードが RPS の場合にだけ有効です。ポートのモードがスタック電源の場合、スタック

	コマンドまたはアクション	目的
		電源コマンドを使用してプライオリティを設定します。 • **priority** *port-priority*：ポートの RPS プライオリティを設定します。指定できる範囲は 1 ～ 9 です。1 が最も高いプライオリティです。デフォルトのプライオリティは XPS ポート番号です。
ステップ 5	**show env xps port**	ポートの XPS 設定を確認します。
ステップ 6	**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。

XPS 電源装置の設定

手順

	コマンドまたはアクション	目的
ステップ 1	**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	**power xps** *switch-number* **supply** {**A** | **B**} **mode** {**rps** | **sp**}	（注） *switch-number* は、1 ～ 9 の値でデータ スタック内のスイッチ番号を表します。 XPS 電源装置のモードを設定します。 • **supply** {**A** | **B**}：設定する電源装置を選択します。左側が電源装置 A（PS1 と表示）で、右側が電源装置 B（PS2）です。 • **mode rps**：接続しているスイッチをバックアップするには、電源装置のモードを RPS に設定します。これは電源装置 A（PS1）のデフォルト設定です。 • **mode sp**：電源スタックに参加するには、電源装置のモードをスタック電源（SP）に設定します。これは

	コマンドまたはアクション	目的
		電源装置 B（PS2）のデフォルト設定です。
ステップ 3	**power xps** *switch-number* **supply** {**A** | **B**} {**on** | **off**}	（注） *switch-number* は、1 ～ 9 の値でデータ スタック内のスイッチ番号を表します。 XPS 電源装置をオンまたはオフに設定します。デフォルトは、2 台ともオンです。
ステップ 4	**end**	特権 EXEC モードに戻ります。
ステップ 5	**show env xps power**	XPS 電源装置のステータスを表示します。

Cisco Expandable Power System 2200 の監視と保守

コマンド	目的
show env xps system	設定したシステムとポートの名前を確認します。
show env xps port	ポートの XPS 設定を確認します。
show env xps power	XPS 電源装置のステータスを表示します。

Cisco Expandable Power System 2200 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。	の *Command Reference (Catalyst 9200 Series Switches)* 「*Interface and Hardware Commands*」の項を参照してください。

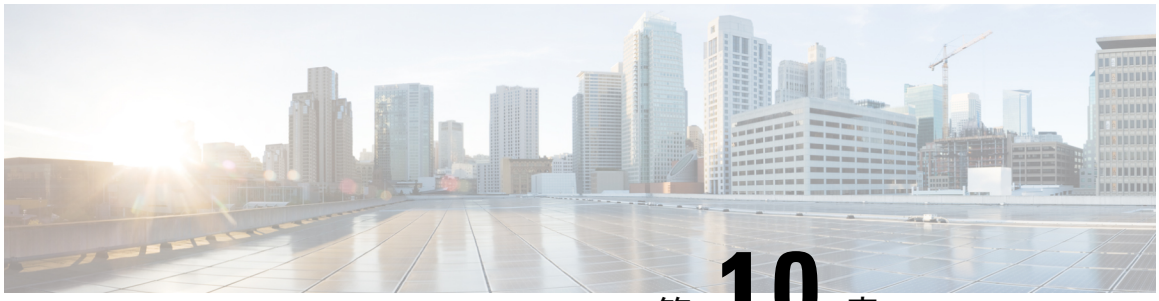
Cisco Expandable Power System 2200 の機能履歴

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 16.9.2	Cisco Expandable Power System（XPS）2200	XPS 2200 は、接続されている装置で電源装置の故障が発生した場合、その装置にバックアップ電力を供給できるスタンドアロン電源システムです。また、Catalyst 電源スタックでは、電源スタック バジェットに追加の電力を供給できます。

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

EEE の設定

EEE の制約事項

Energy Efficient Ethernet（EEE）には次の制限があります。

- EEE の設定を変更すると、デバイスがレイヤ 1 の自動ネゴシエーションを再起動しなければならないため、インターフェイスがリセットされます。
- 受信パスでデータを受け入れる前により長いウェイク アップ時間を必要とするデバイスのリンク層検出プロトコル（LLDP）を有効にする必要がある場合があります。これにより、デバイスは送信リンク パートナーから拡張システムのウェイク アップ時間についてネゴシエーションできます。

EEE について

EEE の概要

Energy Efficient Ethernet (EEE) は、アイドル時間にイーサネット ネットワークの消費電力を減らすように設計された IEEE 802.3az の標準です。

デフォルトの EEE 設定

EEE の設定方法

EEE 対応リンク パートナーに接続されているインターフェイスの EEE を有効または無効にできます。

EEE の有効化または無効化

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`	設定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	**power efficient-ethernet auto** 例： `Device(config-if)# power efficient-ethernet auto`	特定のインターフェイスで EEE を有効にします。EEE が有効になっている場合、デバイスはリンク パートナーに EEE をアドバタイズし、自動ネゴシエートします。
ステップ 4	**no power efficient-ethernet auto** 例： `Device(config-if)# no power efficient-ethernet auto`	指定したインターフェイス上で EEE を無効にします。
ステップ 5	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
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。

EEE の監視

表 *10 : EEE* 設定を表示するコマンド

コマンド	目的
show eee capabilities interface *interface-id*	指定インターフェイスの EEE 機能を表示します。
show eee status interface *interface-id*	指定したインターフェイスの EEE ステータス情報を表示します。
show eee counters interface *interface-id*	指定したインターフェイスの EEE 機能を表示します。

次に、**show eee** コマンドの例を示します。

```
Switch#show eee capabilities interface gigabitEthernet2/0/1
Gi2/0/1
EEE(efficient-ethernet): yes (100-Tx and 1000T auto)
Link Partner : yes (100-Tx and 1000T auto)

ASIC/Interface : EEE Capable/EEE Enabled
```

```
Switch#show eee status interface gigabitEthernet2/0/1
Gi2/0/1 is up
EEE(efficient-ethernet): Operational
Rx LPI Status : Low Power
Tx LPI Status : Low Power
Wake Error Count : 0

ASIC EEE STATUS
Rx LPI Status : Receiving LPI
Tx LPI Status : Transmitting LPI
Link Fault Status : Link Up
Sync Status : Code group synchronization with data stream intact
```

```
Switch#show eee counters interface gigabitEthernet2/0/1

LP Active Tx Time (10us) : 66649648
LP Transitioning Tx : 462
LP Active Rx Time (10us) : 64911682
LP Transitioning Rx : 153
```

EEE の設定例

次に、インターフェイスで EEE を有効にする例を示します。

```
Device# configure terminal
Device(config)# interface gigabitethernet 1/0/1
Device(config-if)# power efficient-ethernet auto
```

次に、インターフェイスで EEE を無効にする例を示します。

```
Device# configure terminal
Device(config)# interface gigabitethernet 1/0/1
Device(config-if)# no power efficient-ethernet auto
```

EEE に関するその他の関連資料

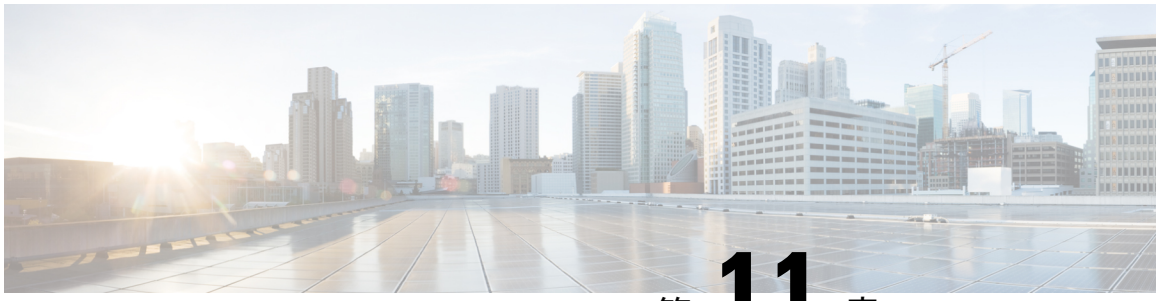
EEE 設定の機能履歴

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 16.9.2	Energy Efficient Ethernet	Energy Efficient Ethernet (EEE) は、アイドル時間にイーサネット ネットワークの消費電力を減らすように設計された IEEE 802.3az の標準です。

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 11 章

Power over Ethernet の設定

Power over Ethernet について

永続的な PoE と 2 イベント分類の設定については、『Network Powered Lighting Configuration Guide, Cisco IOS XE Fuji 16.9.x (Catalyst 9200 Switches)』を参照してください。

次の項では、Power over Ethernet（PoE）、サポートされているプロトコルと標準規格、および電源管理について説明します。

PoE および PoE+ ポート

PoE 対応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デバイスが検出した場合、接続している次のデバイスのいずれかに電力が自動的に供給されます。

- シスコ準規格の受電デバイス（Cisco IP Phone など）
- IEEE 802.3af 準拠の受電デバイス
- IEEE 802.3at 準拠の受電デバイス

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。

サポート対象のプロトコルおよび標準規格

デバイスは、PoE のサポートに次のプロトコルと標準規格を使用します。

- 電力消費を通知する CDP：受電デバイスは、消費している電力量をデバイスに通知します。デバイスはこの電力消費に関するメッセージに応答しません。デバイスは、PoE ポートに電力を供給するか、このポートへの電力を取り除くだけです。
- 高電力装置は、電力ネゴシエーション CDP をサポートしないデバイスでは低電力モードで動作できます。

  Cisco Intelligent Power Management は、電力消費に関して CDP との下位互換性があるため、デバイスは、受信する CDP メッセージに従って応答します。CDP はサードパーティの受電デバイスでサポートされません。このため、デバイスは IEEE 分類を使用して装置の消費電力を判断します。
- IEEE 802.3af：この規格の主な機能は、受電デバイスの検出、電力の管理、切断の検出です。オプションとして受電デバイスの電力分類があります。詳細については、この規格を参照してください。
- IEEE 802.3at：PoE+ 標準では、受電デバイスに供給される最大電力が、1 ポートあたり 15.4 W から 30 W に増えました。

受電デバイスの検出と初期電力割り当て

スイッチは、PoE 対応ポートがシャットダウン状態でなく、PoE が有効になっていて（デフォルト）、接続された装置が AC アダプタから電力供給されていない場合、シスコの準規格受電デバイスまたは IEEE 準拠の受電デバイスを検出します。

装置の検出後、スイッチは、次のように装置のタイプに応じて電力要件を判断します。

- 初期電力割り当ては、受電デバイスが要求する最大電力量です。スイッチは、受電デバイスを検出および電力供給する場合、この電力を最初に割り当てます。スイッチが受電デバイスから CDP メッセージを受信し、受電デバイスが CDP 電力ネゴシエーション メッセージを通じてスイッチと電力レベルをネゴシエートしたときに、初期電力割り当てが調整される場合があります。
- スイッチは検出した IEEE 装置を消費電力クラス内で分類します。スイッチは、電力バジェットに使用可能な電力量に基づいて、ポートに通電できるかどうかを決定します。次の「*IEEE* 電力分類」の表にこれらのレベルを示します。

表 *11 : IEEE* 電力分類

クラス	デバイスから要求される最大電力レベル
0（クラス ステータスは不明）	15.4 W
1	4 W
2	7 W
3	15.4 W

スイッチは電力要求をモニタリングおよび追跡して必要な場合にだけ電力供給を許可します。スイッチはそれ自体の電力バジェット（PoEのデバイスで使用可能な電力量）を追跡します。電力の供給許可または拒否がポートで行われると、スイッチはパワーアカウンティング計算を実行し、電力バジェットを最新に保ちます。

電力がポートに投入された後に、スイッチが CDP を使用して、接続されたシスコ受電デバイスの CDP 固有の電力消費要件を調べます。この要件は、CDP メッセージに基づいて割り当てられる電力量です。スイッチはこれに従って、電力バジェットを調整します。CDP はサードパーティ製のPoEデバイスには適用されません。スイッチは要件を処理して電力の供給または拒否を行います。要求が許可されると、スイッチは電力バジェットを更新します。要求が拒否された場合は、スイッチはポートの電源がオフになるようにし、syslog メッセージを生成して LED を更新します。受電デバイスはより多くの電力を得るために、スイッチとのネゴシエーションを行うこともできます。

PoE+ では、最大 30 W の電力をネゴシエートするめに、受電デバイスが IEEE 802.3at と LLDP 電源をメディア依存インターフェイス（MDI）のタイプ、長さ、および値の説明（TLV）（Power-via-MDI TLV）とともに使用します。シスコの準規格デバイスとシスコの IEEE 受電デバイスは CDP または IEEE 802.3at Power-via-MDI 電力ネゴシエーションメカニズムを使用して最大 30 W の電力レベルを要求できます。

（注） ソフトウェア コンフィギュレーション ガイドおよびコマンド リファレンスでは、CDP 固有の電力消費要件を実際電力消費要件と呼んでいます。

不足電圧、過電圧、過熱、オシレータ障害、または短絡状態による障害をスイッチが検出した場合、ポートへの電力供給をオフにし、syslog メッセージを生成し、電力バジェットと LED を更新します。

電力管理モード

デバイスでは、次の PoE モードがサポートされます。

- **auto**：接続されているデバイスで電力が必要であるかどうかが自動的に検出されます。ポートに接続されている受電デバイスをデバイスが検出し、デバイスに十分な電力がある場合は、電力を供給して電力バジェットを更新し、先着順でポートの電力供給をオンに切り替えます。また、LED を更新します。LED 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。

  すべての受電デバイス用としてデバイスに十分な電力がある場合は、すべての受電デバイスが起動します。デバイスに接続された受電デバイスすべてに対し十分な電力が利用できる場合、すべてのデバイスに電力が供給されます。使用可能なPoEがない場合、または他の装置が電力供給を待機している間に装置の接続が切断されて再接続した場合、どの装置へ電力を供給または拒否されるかが判断できなくなります。

  許可された電力がシステムの電力バジェットを超えている場合、デバイスは電力を拒否し、ポートへの電力がオフになっていることを確認したうえで syslog メッセージを生成し、LEDを更新します。電力供給が拒否された後、デバイスは定期的に電力バジェットを再確認し、継続して電力要求の許可を試みます。

デバイスにより電力を供給されている装置が、さらに壁面コンセントに接続している場合、デバイスは装置に電力を供給し続ける場合があります。このとき、装置がデバイスから受電しているか、AC 電源から受電しているかにかかわらず、デバイスは引き続き装置へ電力を供給していることを報告し続ける場合があります。

受電デバイスが取り外された場合、デバイスは切断を自動的に検出し、ポートから電力を取り除きます。非受電デバイスを接続しても、そのデバイスに障害は発生しません。

ポートで許可される最大ワット数を指定できます。受電デバイスの IEEE クラス最大ワット数が、設定されている最大値より大きい場合、デバイスはそのポートに電力を供給しません。デバイスが受電デバイスに電力を供給する場合でも、受電デバイスが設定された最大値を超える電力を CDP または LLDP メッセージを通じて後から要求すると、デバイスはポートへの電力供給を行いません。その受電デバイスに割り当てられていた電力は、グローバル電力バジェットに送られます。ワット数を指定しない場合、デバイスは最大値の電力を供給します。任意の PoE ポートで **auto** 設定を使用してください。auto モードがデフォルト設定です。

- **static**：デバイスは、受電デバイスが接続されていなくてもポートに電力をあらかじめ割り当て、そのポートで電力が使用できるようにします。デバイスは、設定された最大ワット数をポートに割り当てます。その値は、IEEE クラスまたは受電デバイスからの CDP メッセージによって調節されることはありません。これは、電力があらかじめ割り当てられていることから、最大ワット数以下の電力を使用するすべての受電デバイスが固定ポートに接続されている場合に電力が保証されるためです。ポートはもう先着順方式ではなくなります。

  ただし、受電デバイスの IEEE クラスが最大ワット数を超えると、デバイスは装置に電力を供給しません。受電デバイスが最大ワット数を超える電力を消費していることを CDP メッセージによって知ると、デバイスは受電デバイスをシャットダウンします。

  ワット数を指定しない場合、デバイスは最大数をあらかじめ割り当てます。デバイスは、受電デバイスを検出した場合に限り、ポートに電力を供給します。優先順位が高いインターフェイスには、**static** 設定を使用してください。

- **never**：デバイスは受電デバイスの検出を無効にして、電力が供給されていないデバイスが接続されても、PoE ポートに電力を供給しません。PoE 対応ポートに電力を絶対に適用せず、そのポートをデータ専用ポートにする場合に限り、このモードを使用してください。

ほとんどの場合、デフォルトの設定（自動モード）の動作は適切に行われ、プラグアンドプレイ動作が提供されます。それ以上の設定は必要ありません。ただし、優先順位の高いPoEポートを設定したり、PoEポートをデータ専用にしたり、最大ワット数を指定して高電力受電デバイスをポートで禁止したりする場合は、このタスクを実行します。

電力モニタリングおよび電力ポリシング

リアルタイム電力消費のポリシングを有効にした場合、受電デバイスが最大割り当て量（カットオフ電力値）を超えて電力を消費すると、デバイスはアクションを開始します。

PoEが有効になっている場合、デバイスは受電デバイスのリアルタイムの電力消費を検知します。接続されている受電デバイスのリアルタイム電力消費をデバイスが監視することを、電力

モニタリングまたは電力検知といいます。また、デバイスは電力ポリシング機能を使用して消費電力をポリシングします。

電力モニタリングは、シスコのインテリジェントな電力管理および CDP ベースの消費電力に対して下位互換性があります。電力モニタリングはこれらの機能とともに動作して、PoE ポートが受電デバイスに電力を供給できるようにします。

デバイスは次のようにして、接続されている装置のリアルタイム電力消費を検知します。

1. デバイスは、個々のポートでリアルタイム消費電力をモニタリングします。
2. デバイスは、ピーク時の電力消費を含め、電力消費を記録します。デバイスは CISCO-POWER-ETHERNET-EXT-MIB を介して情報を報告します。
3. 電力ポリシングが有効になっている場合、デバイスはリアルタイムの消費電力を装置に割り当てられた最大電力と比較して、消費電力をポリシングします。最大消費電力は、PoE ポートでカットオフ電力とも呼ばれます。

   デバイスがポートで最大電力割り当てを超える電力を使用すると、デバイスはポートへの電力供給をオフにしたり、またはデバイスの設定に基づいて受電デバイスに電力を供給しながらデバイスが syslog メッセージを生成したり、 LED（ポート LED はオレンジ色で点滅）を更新したりすることができます。デフォルトでは、すべての PoE ポートで消費電力のポリシングは無効になっています。

   PoE の error-disabled ステートからのエラー回復が有効になっている場合、指定の時間の経過後、デバイスは PoE ポートを error-disabled ステートから自動的に回復させます。

   エラー回復が無効になっている場合、**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 コマンドを使用して、手動で PoE ポートを有効にできます。
4. ポリシングが無効になっている場合、受電デバイスが PoE ポートに割り当てられた最大電力より多くの量を消費しても対処されないため、デバイスに悪影響を与える場合があります。

電力消費値

ポートの初期電力割り当ておよび最大電力割り当てを設定することができます。ただし、これらの値は、デバイスが PoE ポートの電力供給をオンまたはオフにするタイミングを指定するために設定する値です。最大電力割り当ては、受電デバイスの実際の電力消費と同じではありません。デバイスによって電力ポリシングに使用される実際のカットオフ電力値は、設定済みの電力値と同等ではありません。

電力ポリシングが有効になっている場合、デバイスは、スイッチポートで受電デバイスの消費電力を超える消費電力ポリシングを行います。最大電力割り当てを手動で設定する場合、スイッチポートと受電デバイス間のケーブルでの電力損失を考慮する必要があります。カットオフ電力とは、受電デバイスの定格消費電力とケーブル上での最悪時の電力損失を合計したものです。

デバイスの PoE が有効になっている場合、電力ポリシングを有効にすることを推奨します。たとえば、クラス 1 デバイスの場合、ポリシングが無効になっており、**power inline auto max 6300** 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カットオフ電力値を設定

すると、PoE ポートに設定される最大電力割り当ては 6.3 W（6300 mW）になります。装置が最大で 6.3 W の電力を必要とする場合、デバイスはポートに接続されている装置に電力を供給します。CDP によるパワーネゴシエーション実施後の値または IEEE 分類値が設定済みカットオフ値を超えると、デバイスは接続されている装置に電力を供給しなくなります。デバイスは PoE ポートで電力供給をオンにした後、受電デバイスのリアルタイム電力消費のポリシングを行わないので、受電デバイスは最大割り当て量を超えて電力を消費できることになり、デバイスと、他の PoE ポートに接続されている受電デバイスに悪影響を及ぼすことがあります。

PoE と UPOE の設定方法

次のタスクでは、PoE と UPOE の設定方法について説明します。

PoE ポートの電力管理モードの設定

（注） PoE 設定を変更するとき、設定中のポートでは電力が低下します。新しい設定、その他の PoE ポートの状態、電力バジェットの状態により、そのポートの電力は再びアップしない場合があります。たとえば、ポート 1 が自動でオンの状態になっていて、そのポートを固定モードに設定するとします。デバイスはポート 1 から電力を取り除き、受電デバイスを検出してポートに電力を再び供給します。ポート 1 が自動でオンの状態になっており、最大ワット数を 10 W に設定した場合、デバイスはポートから電力を取り除き、受電デバイスを再び検出します。デバイスは、受電デバイスがクラス 1、クラス 2、またはシスコ専用受電デバイスのいずれかの場合に、ポートに電力を再び供給します。

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	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet2/0/1`	設定する物理ポートを指定し、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。

	コマンドまたはアクション	目的
ステップ 4	**power inline** {**auto** [**max** *max-wattage*] | **never** | **static** [**max** *max-wattage*] } 例： `Device(config-if)# power inline auto`	ポートのPoEモードを設定します。キーワードの意味は次のとおりです。 • **auto**：受電デバイスの検出を有効にします。十分な電力がある場合は、装置の検出後に PoE ポートに電力を自動的に割り当てます。これがデフォルト設定です。 • **max** *max-wattage*：ポートで許可される電力を制限します。値を指定しない場合は、最大電力が供給されます。 • **never**：デバイス検出を無効にし、ポートへの電力供給を無効にします。 （注） ポートにシスコの受電デバイスが接続されている場合は、**power inline never** コマンドでポートを設定しないでください。問題のあるリンクアップが発生し、ポートが error-disabled ステートになることがあります。 • **static**：受電デバイスの検出を有効にします。デバイスが受電デバイスを検出する前に、ポートへの電力を事前に割り当てます（確保します）。デバイスは、デバイスが接続されていなくてもこのポートに電力を予約し、デバイスの検出時に電力が供給されることを保証します。 デバイスは、自動モードに設定されたポートに電力を割り当てる前に、固定モードに設定されたポートに PoE を割り当てます。
ステップ 5	**end** 例： `Device(config-if)# 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 6	| **module** *switch-number* 例： `Device# show power inline`	デバイスかデバイス スタック、または指定したインターフェイスか指定したスタック メンバーの PoE ステータスを表示します。 **module** *switch-number* キーワードはスタッキング対応のデバイスのみでサポートされます。
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

電力ポリシングの設定

デフォルトでは、デバイスは接続されている受電デバイスの消費電力をリアルタイムでモニタリングします。消費電力に対するポリシングを行うようにデバイスを設定できます。デフォルトではポリシングは無効になります。

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	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet2/0/1`	設定する物理ポートを指定し、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 4	**power inline police** [**action** {**log** | **errdisable**}] 例： `Device(config-if)# power inline police`	ポートでリアルタイム消費電力が最大電力割り当てを超える場合、次のいずれかのアクションを実行するようにデバイスを設定します。

	コマンドまたはアクション	目的
		• **power inline police**：PoE ポートをシャット ダウンし、ポートへの電力供給をオフにし、PoE ポートを error-disabled ステートに移行します。 （注） **errdisable detect cause inline-power** グローバル コンフィギュレーション コマンドを使用すると、PoE error-disabled の原因についてエラー検出を有効にできます。**errdisable recovery cause inline-power interval** *interval* グローバル コンフィギュレーション コマンドを使用すると、PoE error-disabled ステートから回復するためのタイマーを有効にすることもできます。 • **power inline police action errdisable**：リアルタイムの電力消費がポートの最大電力割り当てを超過した場合、ポートへの電力供給をオフにします。 • **power inline police action log**：ポートへの電力供給を継続し、syslog メッセージを生成します。 **action log** キーワードを入力しない場合、デフォルトのアクションによってポートがシャットダウンされ、error-disabled ステートになります。
ステップ 5	**exit** 例： `Device(config-if)# 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	次のいずれかを使用します。 • **errdisable detect cause inline-power** • **errdisable recovery cause inline-power**	（任意）PoE error-disabled ステートからのエラー回復を有効にし、PoE 回復メカニズム変数を設定します。

	コマンドまたはアクション	目的
	• **errdisable recovery interval** *interval* 例： `Device(config)# errdisable detect cause inline-power` `Device(config)# errdisable recovery cause inline-power` `Device(config)# errdisable recovery interval 100`	デフォルトでは、回復間隔は 300 秒です。 **interval** *interval* には、error-disabled ステートから回復する時間を秒単位で指定します。指定できる範囲は 30 ～ 86400 です。
ステップ 7	**exit** 例： `Device(config)# exit`	特権 EXEC モードに戻ります。
ステップ 8	次のいずれかを使用します。 • **show power inline police** • **show errdisable recovery** 例： `Device# show power inline police` `Device# show errdisable recovery`	電力モニタリング ステータスを表示し、エラー回復設定を確認します。
ステップ 9	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

電力ステータスのモニタ

Power over Ethernet 設定をモニタリングおよび確認するには、次の **show** コマンドを使用します。

表 *12 :* 電力ステータスの *show* コマンド

コマンド	目的
show power inline police	電力ポリシングのデータを表示します。

PoE 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に関する完全な構文および使用方法の詳細について。	『*Command Reference Guide*』の「Interface and Hardware Commands」の項を参照してください。

Power over Ethernet 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 16.9.2	Power over Ethernet（PoE）	Power over Ethernet（PoE）では、銅線イーサネットケーブル経由で LAN スイッチングインフラストラクチャがエンドポイント（受電デバイスという）に電力を供給できます。次のタイプのエンドポイントに PoE から電力を供給できます。 • シスコ準規格受電デバイス IEEE 802.3af 準拠の受電デバイス IEEE 802.3at 準拠の受電デバイス

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

無停止型 PoE および高速 POE の設定

無停止型および高速 PoE の制約事項

無停止型および高速 PoE には、次の制限が適用されます。

- 高速 PoE または無停止型 PoE の設定は、エンドポイントを物理的に接続する前に行う必要があります。または、電力を供給しているポートの手動 shut/no-shut を行います。
- ポートへの電力供給は MCU ファームウェアのアップグレード時には中断され、ポートはアップグレード直後にバックアップされます。
- DHCP サーバから割り当てられた IP が設定されていない場合、CREE ライト電力供給デバイス（PD）は定期的にフラップすることがあります。
- PD が LLDP をサポートしていない場合、ユーザはスタティックまたは 2 イベントを設定して、PD 仕様に従って必要な電力を受け取ることができます。

無停止型 POE

無停止型 PoE は、電源装置（PSE）スイッチが再ロード中および起動中であっても、接続されている電源供給を受けるデバイス（PD）へ中断なく電力を提供します。

（注） ポートへの電力供給は MCU ファームウェアのアップグレード時には中断され、ポートはアップグレード直後にバックアップされます。

高速 POE

この機能は、IOS の起動を待たずに電源をオンにします。**poe-ha** が特定のポートで有効な場合、電源障害後の復旧時に、IOS転送が開始されるまでの短期間、スイッチが接続されてるエンドポイントデバイスに電源を供給します。

無停止型および高速 PoE の設定

無停止型および高速 PoE 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 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-id* 例： `Device(config)# interface gigabitethernet 2/0/1`	設定する物理ポートを指定し、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 4	**power inline port perpetual-poe-ha** 例： `Device(config-if)# power inline port perpetual-poe-ha`	無停止型 PoE を設定します。PD デバイスに接続されたポートに無停止型 PoE を設定すると、リロード中に PD デバイスの電源がオンのままになります。
ステップ 5	**power inline port poe-ha** 例： `Device(config-if)# power inline port poe-ha`	高速 PoE を設定します。高速 PoE を設定する場合、スイッチの電源を再投入すると、IOS の起動を待たずに電源に接続してから 50 ー 60 秒以内に PD デバイスの電源がオンになります。

	コマンドまたはアクション	目的
ステップ 6	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。

例：無停止型および高速 PoE の設定

次の例では、スイッチ上で無停止型 PoE を設定にする方法を示します。

```
Device> enable
Device# configure terminal
Device(config)# interface gigabitethernet2/0/1
Device(config-if)# power inline port perpetual-poe-ha
Device(config-if)# end
```

次の例では、スイッチ上で高速 PoE を設定にする方法を示します。

```
Device> enable
Device# configure terminal
Device(config)# interface gigabitethernet2/0/1
Device(config-if)# power inline port poe-ha
Device(config-if)# end
```

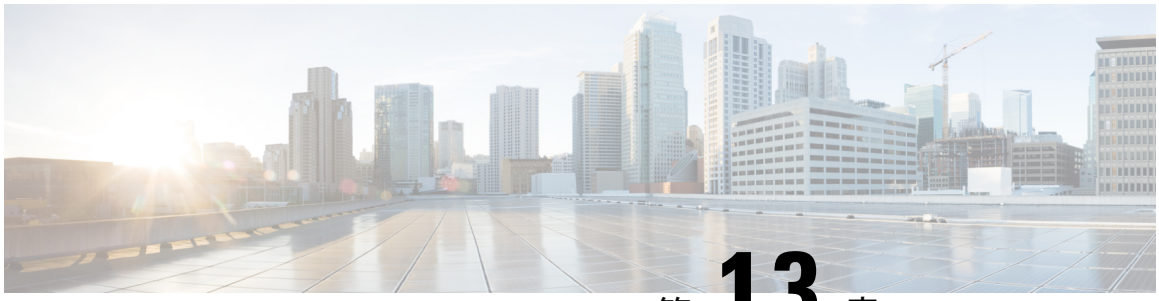
無停止型および高速 PoE の機能情報

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

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

表 *13* : 無停止型および高速 *PoE* の機能情報

機能名	リリース	機能情報
無停止型高速 PoE	Cisco IOS XE Fuji 16.9.2	無停止型 POE は、PSE スイッチが起動している場合でも、接続された PD デバイスへの連続電源を提供します。 高速 PoE は、IOS の起動を待たずに電源をオンにします。



第 13 章

2 イベント分類の設定

2 イベント分類の制約事項

2 イベント分類には次の制約が適用されます。

- 2 イベント分類の設定は、エンドポイントを物理的に接続する前に行っておく必要があります。または、電力を供給しているポートの手動 shut/no-shut を行います。
- ポートへの電力供給は MCU ファームウェアのアップグレード時には中断され、ポートはアップグレード直後にバックアップされます。

2 イベント分類について

クラス 4 デバイスが検出されると、IOS は、CDP または LLDP のネゴシエーションを行うことなく 30W を割り当てます。これは、リンクがアップする前であっても、クラス 4 の電源デバイスは 30W を得ることを意味します。

また、ハードウェア レベルで、PSE は 2 イベント分類を行い、これにより、クラス 4 PD はハードウェアから 30W を供給する PSE の能力を検出し、それ自体を登録することができます。また、CDP/LLDP パケット交換を待つことなく最大 PoE+ レベルまで移動できます。

2 イベントがポートで有効になったら、ポートの遮断または開放を手動で行うか、または PD を再度接続して IEEE 検出を再度開始する必要があります。2 イベント分類がポートで有効になっている場合、クラス 4 デバイスの電力バジェット割り当ては 30W です。その他の場合は 15.4W です。

2 イベント分類の設定

2 イベント分類の設定

2 イベント分類について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 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-id* 例： `Device(config)# interface gigabitethernet2/0/1`	設定する物理ポートを指定し、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 4	**power inline port 2-event** 例： `Device(config-if)# power inline port 2-event`	スイッチで 2 イベント分類を設定します。
ステップ 5	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。

例：2 イベント分類の設定

次に、2 イベント分類を設定する例を示します。

```
Device> enable
Device# configure terminal
Device(config)#  interface gigabitethernet2/0/1
Device(config-if)# power inline port 2-event
Device(config-if)# end
```

2 イベント分類の機能情報

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

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

表 *14 : 2* イベント分類の機能情報

機能名	リリース	機能情報
2 イベント分類	Cisco IOS XE Fuji 16.9.2	クラス 4 デバイスが検出されると、IOS は、CDP または LLDP のネゴシエーションを行うことなく 30W を割り当てます。これは、リンクがアップする前であっても、クラス 4 の電源デバイスは 30W を得ることを意味します。

第 14 章

Auto SmartPorts の設定

Auto SmartPorts の設定の制約事項

Auto SmartPort は Cisco スイッチを検出しますが、イベントトリガーを自動的に呼び出しません。スイッチをマクロにマッピングするには、イベントトリガーを手動で呼び出す必要があります。

no macro auto global processing コマンドは、Auto Smartport のみを無効にします。デバイス分類子を無効にするには、**no device classifier** コマンドを使用します。

Auto SmartPorts に関する情報

Auto SmartPort マクロは、ポートで検出されたデバイス タイプに基づいてポートを動的に設定します。スイッチがポートで新しいデバイスを検出すると、適切な Auto SmartPorts マクロを適用します。ポート上でリンク ダウン イベントが発生した場合、スイッチはそのマクロを削除します。たとえば、ポートに Cisco IP Phone を接続した場合は、Auto SmartPorts により自動的に Cisco IP Phone マクロが適用されます。Cisco IP Phone マクロが適用されると、遅延に影響されやすい音声トラフィックを正しく処理できるように QoS（Quality Of Service）、セキュリティ機能、および専用の音声 VLAN がイネーブルになります。

Auto SmartPorts は、イベントトリガーを使用して、マクロにデバイスをマッピングします。最も一般的なイベントトリガーは、接続されているデバイスから受信した Cisco Discovery Protocol

（CDP）メッセージに基づいています。デバイス（Cisco IP Phone、Cisco ワイヤレスアクセスポイント、または Cisco ルータ）の検出は、そのデバイスのイベントトリガーを呼び出します。

Link Layer Discovery Protocol（LLDP）は、CDP をサポートしないデバイスを検出するために使用されます。イベント トリガーとして使用される他のメカニズムには、802.1X 認証結果と学習した MAC アドレスなどがあります。

主に CDP および LLDP メッセージと MAC アドレスに基づいて、さまざまなデバイス用にシステムの組み込みイベント トリガーがあります。これらのトリガーは、Auto SmartPort が有効になっている限り有効になっています。

プロファイルとデバイス用のユーザ定義のトリガー グループを設定できます。トリガー グループ名を使用してユーザ定義マクロを関連付けます。

Auto SmartPorts マクロ

Auto SmartPort マクロは CLI コマンドのグループです。ポートのデバイスが検出されると、デバイスにマクロが適用されます。システムの組み込みマクロはさまざまなデバイスに存在し、デフォルトでは、システムの組み込みのトリガーは、対応する組み込みマクロにマッピングされます。必要に応じて、組み込みのトリガーまたはマクロのマッピングを変更できます。

マクロは、基本的に、リンク ステータスに基づいて、インターフェイスの CLI のセットを適用または削除します。マクロでは、リンク ステータスがチェックされます。リンクがアップ ステータスの場合は、CLI のセットが適用されます。リンクがダウンしている場合、セットが削除されます（CLI の no 形式が適用されます）。CLI のセットを適用するマクロの部分は、マクロと呼びます。CLI を削除する部分（CLI の no 形式）は、アンチマクロと呼びます。

デバイスが Auto SmartPort に接続されている場合に、点灯しているエンドポイントとして分類されると、イベントトリガー **CISCO_LIGHT_EVENT** が呼び出され、マクロ **CISCO_LIGHT_AUTO_SMARTPORT** が実行されます。

CISCO_LIGHT_AUTO_SMARTPORTによって実行されるコマンド

マクロが実行されると、スイッチで一連のコマンドが実行されます。

マクロ **CISCO_LIGHT_AUTO_SMARTPORT** を実行することで実行されるコマンドは、次のとおりです。

- switchport mode access
- switchport port-security violation restrict
- switchport port-security mac-address sticky
- switchport port-security

- power inline port poe-ha
- storm-control broadcast level 50.00
- storm-control multicast level 50.00
- storm-control unicast level 50.00
- spanning-tree portfast
- spanning-tree bpduguard enable

Auto SmartPort の有効化

（注） Auto SmartPort はデフォルトで無効になっています。

特定のポートの Auto SmartPorts マクロをディセーブルにするには、Auto SmartPort をグローバルにイネーブルにする前に、**no macro auto global processing** インターフェイス コマンドを使用します。

Auto SmartPort をグローバルにイネーブルにするには、**macro auto global processing** 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。

Auto SmartPorts をイネーブルにするには、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 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	**device classifier** 例： `Device(config)# device classifier`	デバイスの分類子を有効にします。 デバイス分類子を無効にするには、**no device classifier** 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	**macro auto global processing** 例： `Device(config)# macro auto global processing`	スイッチの Auto SmartPorts をグローバルにイネーブルにします。 Auto SmartPort をグローバルに無効にするには、**no macro auto global processing** コマンドを使用します。
ステップ 5	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

イベントトリガーと組み込みマクロ間のマッピングの設定

（注） Cisco スイッチが Auto SmartPort に接続されている場合は、このタスクを実行する必要があります。

組み込みマクロにイベントトリガーをマッピングするには、次の作業を行います。

始める前に

auto smartport マクロをグローバルに有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。 • 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 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	**macro auto execute** *event trigger* **builtin** *built-in macro name* 例： `Device(config)# macro auto execute CISCO_SWITCH_EVENT builtin CISCO_SWITCH_AUTO_SMARTPORT`	ユーザ定義のイベント トリガーとマクロ名を指定します。このアクションは、イベントトリガーから組み込み Auto Smartport マクロへのマッピングを設定します。
ステップ 4	**macro auto trigger** *event trigger* 例： `Device(config)# macro auto trigger CISCO_SWITCH_EVENT`	ユーザ定義イベントトリガーを呼び出します。
ステップ 5	**device** *device_ID* 例： `Device(config)# device cisco WS-C3560CX-8PT-S`	イベントトリガーをデバイス ID と照合します。
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 shell triggers** 例： `Device# show shell triggers`	スイッチ上のイベント トリガーを表示します。
ステップ 8	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 9	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

例：Auto SmartPorts の有効化

この例では、Auto SmartPort を有効にする方法を示します。

```
Device> enable
Device# configure terminal
Device(config)# device classifier
Device(config)# macro auto global processing
Device(config)# end
```

例：イベントトリガーと組み込みマクロ間のマッピングの設定

この例では、イベントトリガーと組み込みマクロ間のマッピングを設定する方法を示します。

```
Device> enable
Device# configure terminal
Device(config)# macro auto execute CISCO_SWITCH_EVENT builtin CISCO_SWITCH_AUTO_SMARTPORT
Device(config)# macro auto trigger CISCO_SWITCH_EVENT
Device(config)# device cisco WS-C3560CX-8PT-S
Device(config)# end
```

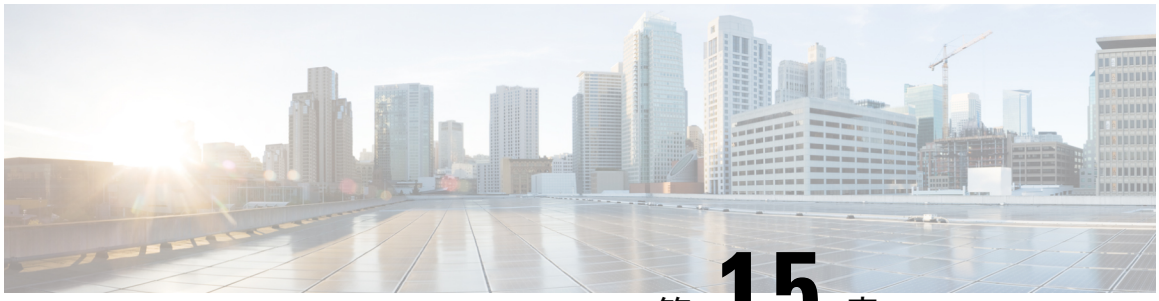
Auto SmartPorts の機能情報

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

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

表 *15 : Auto SmartPorts* の機能情報

機能名	リリース	機能情報
自動 SmartPorts	Cisco IOS XE Fuji 16.9.2	Auto SmartPort マクロは、ポートで検出されたデバイス タイプに基づいてポートを動的に設定します。スイッチがポートで新しいデバイスを検出すると、適切な Auto SmartPorts マクロを適用します。



第 15 章

COAP プロキシ サーバの設定

COAP プロキシ サーバの制約事項

次の制約事項は、COAP プロキシ サーバに適用されます。

- スイッチは、ipv6 ブロードキャスト（CSCuw26467）を使用する CoAP クライアントとして自身をアドバタイズできません。
- 監視のサポートは実装されていません。
- Blockwise 要求はサポートされていません。シスコは、block-wise 応答を処理し、block-wise 応答を生成できます。
- DTLS サポートは、RawPublicKey および証明書ベースのモードに対してのみ有効です。
- スイッチは、DTLS クライアントとして動作しません。DTLS はエンドポイントに対してのみ。
- エンドポイントは、CBOR ペイロードを処理し、応答すると想定されています。
- クライアント側要求は、JSON であると想定されています。
- IPv6 ブロードキャストの問題により、スイッチは IPv6 として他のリソース ディレクトリに自身をアドバタイズすることはできません。

COAP プロキシ サーバについて

COAP プロトコルは、制限されたデバイスで使用できるように設計されています。HTTP が情報にアクセスする際にサーバ上で動作するのと同じ方法で、COAPは制限されたデバイス上で動作します。

COAP と HTTP の比較を次に示します。

- Web サーバの場合、プロトコル は **HTTP**、トランスポートは **TCP**、転送される最も一般的な情報の形式は **HTML** です。
- 制約付きデバイスの場合、プロトコルは **COAP**、トランスポートは **UDP**、一般的な情報の形式は **JSON/link-format/CBOR** です。

COAP によって、HTTP の場合と同様に **GET**/**POST** メタファーと RESTful API を使用してデバイスにアクセスし、管理する手段が提供されます。

COAP プロキシ サーバの設定方法

COAP プロキシ サーバを設定するには、コンフィギュレーション モードで COAP プロキシと COAP エンドポイントを設定できます。

コマンドは **coap [proxy | endpoints]** です。

COAP プロキシの設定

スイッチで COAP プロキシを開始または停止するには、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 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	**coap proxy** 例：	COAP プロキシ サブモードを開始します。

	コマンドまたはアクション	目的
	`Device(config)# coap proxy`	（注） coap proxy を停止して、coap proxy の下にあるすべての設定を削除するには、**no coap proxy** コマンドを使用します。
ステップ 4	**security** [**none** [[**ipv4** | **ipv6**] {*ip-address ip-mask/prefix*} | **list** {*ipv4-list name* | *ipv6-list-name*}] | **dtls** [**id-trustpoint** {*identity-trustpoint label*}] [**verification-trustpoint** {*verification-trustpoint*} | [**ipv4** | **ipv6** {*ip-address ip-mask/prefix*}] | **list** {*ipv4-list name* | *ipv6-list-name*}]] 例： `Device(config-coap-proxy)# security none ipv4 1.1.0.0 255.255.0.0`	暗号化タイプを引数と見なします。サポートされる 2 つのセキュリティ モードは **none** と **dtls** です。 • none：そのポートにセキュリティがないことを示します。 **security none** を使用すると、最大 5 つの IPv4 アドレスと最大 5 つの IPv6 アドレスを関連付けることができます。 • dtls：DTLS セキュリティは、オプションである RSA トラストポイントと検証トラストポイントを要します。検証トラストポイントがないと、通常の公開キー交換が行われます。 **security dtls** を使用すると、最大 5 つの IPv4 アドレスと最大 5 つの IPv6 アドレスを関連付けることができます。 （注） coap proxy のすべてのセキュリティ設定を削除するには、**no security** コマンドを使用します。
ステップ 5	**max-endpoints** {*number*} 例： `Device(config-coap-proxy)#max-endpoints 10`	（任意）スイッチで学習できるエンドポイントの最大数を指定します。デフォルト値は 10 です。指定できる範囲は 1 ～ 500 です。 （注） coap proxy に設定されたすべての最大エンドポイントを削除するには、**no max-endpoints** 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 6	**port-unsecure** {*port-num*} 例： `Device(config-coap-proxy)#port-unsecure 5683`	（任意）デフォルト 5683 以外のポートを設定します。指定できる範囲は 1 ～ 65000 です。 （注） coap proxy のすべてのポート設定を削除するには、**no port-unsecure** コマンドを使用します。
ステップ 7	**port-dtls** {*port-num*} 例： `Device(config-coap-proxy)#port-dtls 5864`	（任意）デフォルト 5684 以外のポートを設定します。 （注） coap proxy のすべて DTLS のポート設定を削除するには、**no port-dtls** コマンドを使用します。
ステップ 8	**resource-directory** [ipv4 | ipv6] {*ip-address*}] 例： `Device(config-coap-proxy)#resource-directory ipv4 192.168.1.1`	スイッチが COAP クライアントとして動作できるユニキャスト アップストリーム リソースのディレクトリ サーバを設定します。 **resource-directory** を使用すると、最大 5 つの IPv4 アドレスと最大 5 つの IPv6 アドレスを設定できます。 （注） coap proxy のすべてのリソースディレクトリ設定を削除するには、**no resource-directory** コマンドを使用します。
ステップ 9	**list** [**ipv4 | ipv6**] {*list-name*} 例： `Device(config-coap-proxy)#list ipv4 trial_list`	（任意）ライトとリソースを学習できる IP アドレス範囲を制限します。上記の **security** [**none** | **dtls**] コマンドオプションで使用する、IP アドレス/マスクの名前付きリストを作成します。 **list** を使用して、IPv4 または IPv6 に関係なく、最大 5 つの IP リストを設定できます。IP リストにつき最大 5 つの IP アドレスを設定できます。 （注） COAP プロキシサーバの IP リストを削除するには、 **no list** [**ipv4 | ipv6**] {*list-name*} 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 10	**start** 例： `Device(config-coap-proxy)#start`	このスイッチで COAP プロキシを開始します。
ステップ 11	**stop** 例： `Device(config-coap-proxy)#stop`	このスイッチで COAP プロキシを停止します。
ステップ 12	**exit** 例： `Device(config-coap-proxy)# exit`	COAP プロキシ サブモードを終了します。
ステップ 13	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

COAP エンドポイントの設定

複数の IPv4/IPv6 スタティック エンドポイントをサポートするように COAP プロキシを設定するには、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 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	**coap endpoint** [**ipv4** | **ipv6**] {*ip-address*} 例： `Device(config)#coap endpoint ipv4 1.1.1.1` `Device(config)#coap endpoint ipv6 2001::1`	スイッチ上でスタティック エンドポイントを設定します。 • **ipv4** ：IPv4 スタティック エンドポイントを設定します。 • **ipv6** ：IPv6 スタティック エンドポイントを設定します。 （注） エンドポイントで coap proxy を停止するには、**no coap endpoint** [**ipv4** |**ipv6**] {*ip-address*} コマンドを使用します。
ステップ 4	**exit** 例： `Device(config-coap-endpoint)# exit`	COAP エンドポイント サブモードを終了します。
ステップ 5	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

COAP プロキシサーバの設定例

例：COAP プロキシ サーバの設定

次の例に、最大 10 のエンドポイントをサポートするようにポート番号 5683 を設定する方法を示します。

```
#coap proxy security none ipv4 2.2.2.2 255.255.255.0 port 5683 max-endpoints 10

---------------------------------------------------------------------------------------------
```

次の例に、セキュリティ設定がされて**いない** *ipv4 1.1.0.0 255.255.0.0* に COAP プロキシを設定する方法を示します。

```
Device(config-coap-proxy)# security ?
  dtls  dtls
  none  no security

Device(config-coap-proxy)#security none ?
  ipv4    IP address range on which to learn lights
```

```
  ipv6    IPv6 address range on which to learn lights
  list    IP address range on which to learn lights

Device(config-coap-proxy)#security none ipv4 ?
  A.B.C.D  {/nn || A.B.C.D}  IP address range on which to learn lights

Device(config-coap-proxy)#security none ipv4 1.1.0.0 255.255.0.0

------------------------------------------------------------------------------------------
```

次の例に、**dtls id trustpoint** セキュリティ設定がされている *ipv4 1.1.0.0 255.255.0.0* に COAP プロキシを設定する方法を示します。

```
Device(config-coap-proxy)#security dtls ?
  id-trustpoint DTLS RSA and X.509 Trustpoint Labels
  ipv4 IP address range on which to learn lights
  ipv6 IPv6 address range on which to learn lights
  list IP address range on which to learn lights

Device(config-coap-proxy)#security dtls id-trustpoint ?
  WORD  Identity TrustPoint Label

Device(config-coap-proxy)#security dtls id-trustpoint RSA-TRUSTPOINT ?
  verification-trustpoint  Certificate Verification Label
  <cr>

Device(config-coap-proxy)#security dtls id-trustpoint RSA-TRUSTPOINT

Device(config-coap-proxy)#security dtls ?
  id-trustpoint DTLS RSA and X.509 Trustpoint Labels
  ipv4 IP address range on which to learn lights
  ipv6 IPv6 address range on which to learn lights
  list IP address range on which to learn lights

Device(config-coap-proxy)# security dtls ipv4 1.1.0.0 255.255.0.0
```

（注） **ipv4 / ipv6 / list** を設定するには、**id-trustpoint** と（任意）**verification-trustpoint** を事前に設定しておく必要があります。設定していない場合はエラーが表示されます。

次の例に、トラストポイントを設定する方法を示します。これは、**id trustpoint** 設定の COAP **security dtls** の前提条件です。

```
ip domain-name myDomain
crypto key generate rsa general-keys exportable label MyLabel modulus 2048

Device(config)#crypto pki trustpoint MY_TRUSTPOINT
Device(ca-trustpoint)#rsakeypair MyLabel 2048
Device(ca-trustpoint)#enrollment selfsigned
Device(ca-trustpoint)#exit

Device(config)#crypto pki enroll MY_TRUSTPOINT
% Include the router serial number in the subject name? [yes/no]: no
% Include an IP address in the subject name? [no]: no
```

```
Generate Self Signed Router Certificate? [yes/no]: yes
```

次の例に、**dtls verification trustpoint** によって *ipv4 1.1.0.0 255.255.0.0* に COAP プロキシを設定する方法を示します（証明書または検証トラストポイントによる DTLS）。

```
Device(config-coap-proxy)#security dtls ?
  id-trustpoint DTLS RSA and X.509 Trustpoint Labels
  ipv4 IP address range on which to learn lights
  ipv6 IPv6 address range on which to learn lights
  list IP address range on which to learn lights

Device(config-coap-proxy)#security dtls id-trustpoint ?
  WORD  Identity TrustPoint Label

Device(config-coap-proxy)#security dtls id-trustpoint RSA-TRUSTPOINT ?
  verification-trustpoint  Certificate Verification Label
  <cr>

Device(config-coap-proxy)#security dtls id-trustpoint RSA-TRUSTPOINT
verification-trustpoint ?
  WORD  Identity TrustPoint Label

Device(config-coap-proxy)#security dtls id-trustpoint RSA-TRUSTPOINT
verification-trustpoint CA-TRUSTPOINT ?
  <cr>
```

次の例に、検証トラストポイントを設定する方法を示します。これは、**verification trustpoint** 設定の COAP **security dtls** の前提条件です。

```
Device(config)#crypto pki import CA-TRUSTPOINT pkcs12 flash:hostA.p12 password cisco123
% Importing pkcs12...
Source filename [hostA.p12]?
Reading file from flash:hostA.p12
CRYPTO_PKI: Imported PKCS12 file successfully.
```

次の例に、セキュリティ [none | dtls] コマンド オプションで使用する、trial-list という名前のリストを作成する方法を示します。

```
Device(config-coap-proxy)#list ipv4 trial_list
Device(config-coap-proxy-iplist)#1.1.0.0 255.255.255.0
Device(config-coap-proxy-iplist)#2.2.0.0 255.255.255.0
Device(config-coap-proxy-iplist)#3.3.0.0 255.255.255.0
Device(config-coap-proxy-iplist)#exit
Device(config-coap-proxy)#security none list trial_list
```

次の例に、coap プロキシ サブ モードで使用できるすべての拒否コマンドを示します。

```
Device(config-coap-proxy)#no ?
  ip-list             Configure IP-List
  max-endpoints       maximum number of endpoints supported
```

```
  port-unsecure       Specify a port number to use
  port-dtls           Specify a dtls-port number to use
  resource-discovery  Resource Discovery Server
  security            CoAP Security features
```

次の例に、coap プロキシで複数の IPv4/IPv6 スタティック エンドポイントを設定する方法を示します。

```
Device(config)# coap endpoint ipv4 1.1.1.1
Device(config)# coap endpoint ipv4 2.1.1.1
Device(config)# coap endpoint ipv6 2001::1
```

次の例に、COAP プロトコルの詳細を表示する方法を示します。

```
Device#show coap version
CoAP version 1.0.0
RFC 7252
```

```
Device#show coap resources
Link format data =
</>
</1.1.1.6/cisco/context>
</1.1.1.6/cisco/actuator>
</1.1.1.6/cisco/sensor>
</1.1.1.6/cisco/lldp>
</1.1.1.5/cisco/context>
</1.1.1.5/cisco/actuator>
</1.1.1.5/cisco/sensor>
</1.1.1.5/cisco/lldp>
</cisco/flood>
</cisco/context>
</cisco/showtech>
</cisco/lldp>
```

```
Device#show coap globals
Coap System Timer Values :
   Discovery  : 120 sec
   Cache Exp  : 5 sec
   Keep Alive : 120 sec
   Client DB  : 60 sec
   Query Queue: 500 ms
   Ack delay  : 500 ms
   Timeout    : 5 sec

Max Endpoints      : 10
Resource Disc Mode : POST
```

```
Device#show coap stats
Coap Stats :
Endpoints : 2
Requests : 20
Ext Queries : 0
```

```
Device#show coap endpoints
List of all endpoints :

Code : D - Discovered , N - New
#    Status   Age(s)  LastWKC(s)  IP
------------------------------------------------------------------------------
1    D        10      94          1.1.1.6
2    D        6       34          1.1.1.5

Endpoints - Total : 2 Discovered : 2 New : 0

-------------------------------------------------------------------------------------------

Device#show coap dtls-endpoints
#     Index State  String State     Value   Port IP
-----------------------------------------------------------------
1     3     SSLOK       3           48969   20.1.1.30
2     2     SSLOK       3           53430   20.1.1.31
3     4     SSLOK       3           54133   20.1.1.32
4     7     SSLOK       3           48236   20.1.1.33


-------------------------------------------------------------------------------------------
```

次の例に、COAP プロトコルのデバッグに使用できるすべてのオプションを示します。

```
Device#debug coap ?
all       Debug CoAP all
database  Debug CoAP Database
errors    Debug CoAP errors
events    Debug CoAP events
packet    Debug CoAP packet
trace     Debug CoAP Trace
warnings  Debug CoAP warnings
```

COAP プロキシ サーバのモニタリング

COAP プロトコルの詳細を表示するには、次の表のコマンドを使用します。

表 *16 : COAP* 固有のデータを表示するコマンド

show coap version	IOS COAP バージョンと RFC 情報を表示します。
show coap resources	スイッチのリソースと、スイッチが学習したリソースを表示します。
show coap endpoints	検出され、学習されたエンドポイントを表示します。
show coap globals	タイマー値とエンドポイント値を表示します。
show coap stats	エンドポイント、要求、および外部クエリのメッセージ数を表示します。
show coap dtls-endpoints	dtls エンドポイントのステータスを表示します。

表 *17 : COAP* コマンドをクリアするコマンド

clear coap database	スイッチで学習された COAP、およびエンドポイント情報の内部データベースをクリアします。

COAP プロトコルをデバッグするには、次の表のコマンドを使用します。

表 *18 : COAP* プロトコルをデバッグするコマンド

debug coap database	COAP データベース出力をデバッグします。
debug coap errors	COAP エラー出力をデバッグします。
debug coap events	COAP イベント出力をデバッグします。
debug coap packets	COAP パケット出力をデバッグします。
debug coap trace	COAP トレース出力をデバッグします。
debug coap warnings	COAP 警告出力をデバッグします。
debug coap all	すべての COAP 出力をデバッグします。

（注） デバッグを無効にする場合は、コマンドの前に「**no**」キーワードを追加します。

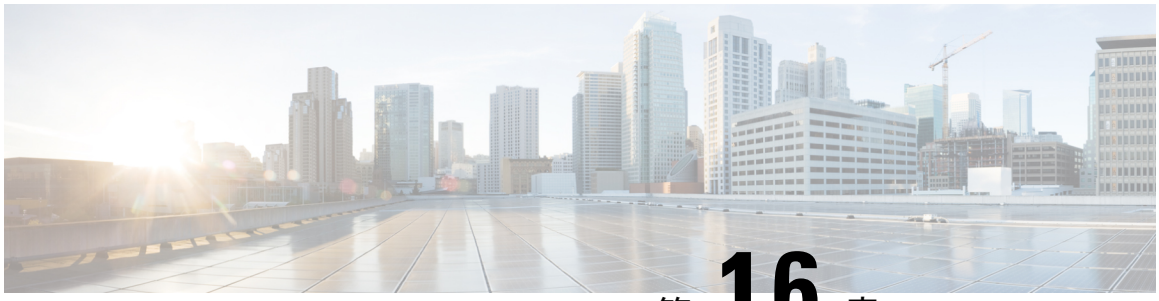
COAP の機能情報

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

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

表 *19 : COAP* の機能情報

機能名	リリース	機能情報
COAP	Cisco IOS XE Fuji 16.9.2	COAP プロトコルは、制限されたデバイスで使用できるように設計されています。HTTP が情報にアクセスする際にサーバ上で動作するのと同じ方法で、COAP は制限されたデバイス上で動作します。



第 16 章

USB 3.0 SSD の設定

USB 3.0 SSD に関する情報

USB 3.0 SSD

USB 3.0 SSD は、アプリケーションをホストするための追加の 120 GB ストレージを提供します。アプリケーションはカーネル仮想マシン（KVM）、Linux Containers（LXC）、または Docker コンテナでホストできます。ストレージドライブを使用して、パケットキャプチャ、オペレーティングシステムによって生成されたトレースログ、およびサードパーティアプリケーションを保存することもできます。USB 3.0 SSD は、汎用ストレージデバイスとして、およびアプリケーションホスティングデバイスとして同時に使用できます。Cisco USB ドライブのみを使用する必要があります。シスコ以外の USB ドライブはサポートされていません。

（注） USB 3.0 SSD は、イメージのブート、イメージの緊急インストール、または（ソフトウェア メンテナンス アップデート（SMU または **install** ））コマンドを使用した内部フラッシュのアップグレードには使用できません。USB 3.0 SSD のブートローダーサポートは使用できません。

USB 3.0 SSD は、ドライブのヘルスモニタリング用に Self-Monitoring、Analysis and Reporting Technology（SMART）機能が有効になっています。S.M.A.R.T の目的は、ドライブの信頼性のモニタ、ドライブ障害の予測、さまざまなタイプのドライブセルフテストを実行することです。SMART Disk Monitoring Daemon（smartd）は、USB 3.0 SSD を挿入した直後に有効になり、/crashinfo/tracelogs/smart_errors.log に警告とエラーのロギングを開始します。これらの警告

とエラーは、コンソールにも表示されます。USB 3.0 SSD を取り外すと、smartd は動作を停止します。

USB 3.0 SSD は、柔軟なストレージ構成を提供する Field Replaceable Unit（FRU）としてサポートされています。最初に PC で SSD を使用する場合、USB 3.0 SSD のデフォルトパーティションが、すべてのファイルシステムをサポートする PC によって作成されます。スイッチで SSD を最初に使用する場合、EXT4 ファイルシステムをサポートするために 120 GB の 1 つのパーティションが作成されます。

USB 3.0 SSD のファイルシステム

USB 3.0 SSD は raw デバイスとして出荷されます。デバイスが起動すると、Cisco IOS ソフトウェアは EXT4 をデフォルトのファイルシステムとしてパーティションを作成します。ただし、デバイスは、EXT2、EXT3、EXT4 などのすべての EXT ベースのファイルシステムをサポートします。VFAT、NTFS、LVM などの非 EXT ベースのファイルシステムはサポートされていません。

ドライブでは、次のファイルシステム操作がサポートされています。

- 読み取り
- 書き込み
- Delete
- Copy
- 書式

USB 3.0 SSD でのパスワード認証

不正アクセスからドライブを保護するには、ユーザパスワードを設定して USB 3.0 SSD のセキュリティを有効にする必要があります。USB 3.0 SSD は、次のセキュリティ状態をサポートします。

- セキュリティ無効：ユーザパスワードがドライブに設定されていません。これは、新しいドライブのデフォルトであるアウトオブボックス状態です。
- セキュリティ有効：ユーザパスワードがドライブに設定されています。
- ロック済み：セキュリティは有効で、ドライブにアクセスできません。
- ロック解除済み：セキュリティは有効または無効ですが、ドライブはアクセス可能です。

CLI およびプログラム可能な NETCONF/YANG 方式を使用してパスワード認証を設定できます。

USB 3.0 SSD の設定方法

USB 3.0 SSD のフォーマット

EXT ファイルシステムまたはドライブ全体をフォーマットするには、**format usbflash1:**{**ext2** | **ext3** | **ext4** | **secure**} コマンドを使用します。

デバイススタックの USB 3.0 SSD ドライブをフォーマットするには、**format usbflash1**-*switch_num*:{**ext2** | **ext3** | **ext4** | **secure**} コマンドを使用します。

スイッチまたはスイッチスタックからの USB 3.0 SSD のマウント解除

スイッチまたはスイッチスタックから USB 3.0 SSD を安全に取り外すには、特権 EXEC モードで **hw-module switch** <*switch_num*> **usbflash1 unmount** コマンドを使用します。このコマンドは、挿入時に作成されたファイルシステムをマウント解除し、システムに保留中の読み取りまたは書き込み操作があれば完了し、スイッチからドライブを安全に取り外すように通知します。

```
Device#hw-module switch 1 usbflash1 unmount

     *Jan  5 22:21:32.723: %IOSXE-0-PLATFORM: Switch 1 R0/0: SSD_UNMOUNT_LOG: usbflash1:
 has been unmounted. All the usbflash1 entries in    IOS will now be cleared until the
SSD is plugged back into the switch.
      *Jan  5 22:21:32.729: %IOSD_INFRA-6-IFS_DEVICE_OIR: Device usbflash1 removed
```

このコマンドを実行すると、USB にアクセスできなくなります。USB を再度使用するには、スイッチに再度挿入します。

USB を挿入せずにスイッチまたはスイッチスタックで **hw-module switch** <*switch_num*> **usbflash1 unmount** コマンドを実行すると、次のエラーメッセージが表示されます。

```
Device#hw-module switch 1 usbflash1 unmount

*Jun 20 22:50:40.321:
ERROR: USB Not Present in this Slot 1
```

USB 3.0 SSD でのパスワードセキュリティの有効化

パスワード認証機能を使用すると、USB 3.0 SSD のセキュリティを設定して、不正アクセスや関連するリスクからドライブを保護できます。USB3.0 SSD のセキュリティを有効にするには、次の手順に従ってドライブにパスワードを設定します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例：	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	**hw-module switch** *switch-number* **usbflash1 security enable password** *usb-password* 例： `Device#hw-module switch 1 usbflash1 security enable password 1234`	USB 3.0 SSD でユーザ定義のパスワードを設定します。 （注） パスワードセキュリティは、USB の活性挿抜（OIR）またはスイッチのリロード後にのみ有効になります。

USB の活性挿抜（OIR）またはスイッチのリロード後、USB は *Enabled and Locked* 状態になります。USB のロックを解除して USB にアクセスするには、このタスクで作成した USB 3.0 SSD パスワードを使用するようにスイッチを設定する必要があります。

次のタスク

スイッチの USB 3.0 SSD パスワードを設定するには、スイッチでの USB 3.0 SSD パスワードの設定 （140 ページ）を参照してください。

スイッチでの USB 3.0 SSD パスワードの設定

スイッチを使用してパスワードで保護された USB 3.0 SSD にアクセスするには、スイッチで同じ USB 3.0 SSD パスワードを設定する必要があります。ドライブのスイッチリセットまたは OIR 後、USB 3.0 SSD はロック状態になります。ドライブのロックを解除してアクセスするために、スイッチに保存されている USB 3.0 SSD パスワードを入力するように求められます。この手順では、パスワードをタイプ 6 暗号化形式でスイッチの実行コンフィギュレーションに保存します。

暗号化事前共有キー機能を使用すると、コマンドライン インターフェイス（CLI）から、プレーンテキストのパスワードをタイプ 6 形式で NVRAM へセキュアに保存できます。タイプ 6 パスワードは暗号化されます。暗号化されたパスワード自体を、確認したり取得したりすることは可能ですが、それを復号化して実際のパスワードを特定することは困難です。

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 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	（任意） **key config-key password-encrypt** *password* 例： `Device(config)#key config-key password-encrypt 123456789`	スイッチのマスターキーを設定します。このコマンドを使用して設定されたパスワードは、スイッチ内のその他すべてのキーを暗号化するマスター暗号キーとして使用されます。 （注） スイッチにマスターキーがすでに設定されている場合は、この手順をスキップします。
ステップ 4	［**no**］ **hw-module switch** *switch-number* **usbflash1-password** *usb-password* 例： `Device(config)#hw-module switch 1 usbflash1-password 1234`	（注） セキュリティを有効にするために、パスワードが USB 3.0 SSD で設定したパスワードと一致することを確認します。 タイプ 6 暗号化を使用してパスワードを内部的に暗号化します。 コマンドの **no** 形式を使用して、スイッチの実行コンフィギュレーションから USB 3.0 SSD パスワードを削除します。
ステップ 5	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。

USB 3.0 SSD のロック解除

USB 3.0 SSD のロックを解除するには、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）。

	コマンドまたはアクション	目的
ステップ 2	**hw-module switch** *switch-number* **usbflash1 security unlock password** *usb-password* 例： `Device#hw-module switch 1 usbflash1 security unlock password 1234`	ドライブのロックを解除し、一時的にアクセスできるようにします。ドライブでパスワードセキュリティが有効になっていることに注意してください。ドライブを他のスイッチに挿入すると、ドライブはロックされた状態になります。

USB 3.0 SSD でのパスワードセキュリティの無効化

セキュリティを無効にする、または USB 3.0 SSD に設定されているパスワードを変更するには、次の手順に従います。

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	**hw-module switch** *switch-number* **usbflash1 security disable password** *usb-password* 例： `Device #hw-module switch 1 usbflash1 security disable password 1234`	USB 3.0 SSD のセキュリティを無効にし、ドライブにアクセスできるようにします。変更を有効にするために、スイッチをリロードしたり、ドライブの OIR を実行したりする必要はありません。 （注） スイッチスタックで、USB 3.0 SSD を挿入したスイッチのスイッチ番号を入力します。

USB 3.0 SSD のモニタリング

USB 3.0 SSD の格納ファイルを操作する前に、その格納ファイルを確認できます。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。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 *20 :* ファイルシステム上のファイルを表示するコマンド

コマンド名	説明
dir usbflash1:	アクティブスイッチ上の USB フラッシュファイルシステム上のファイルのリストを表示します。 スタックのスタンバイスイッチまたは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、**usbflash1**–*n* を使用します（*n* はスタンバイスイッチ番号またはスタックメンバ番号です）。
dir usbflash1-*switch_num***:**	スタックセットアップのファイルシステム上のファイルのリストを表示します。
dir stby-usbflash1:	スタックセットアップのスタンバイスイッチのファイルシステム上のファイルのリストを表示します。
show usbflash1: filesystem	ファイルシステムに関する詳細情報を表示します。
show inventory	USB ハードウェアの物理インベントリ情報を表示します。 複数のスイッチオーバー後、**show inventory** コマンドの出力には、アクティブスイッチの USB フラッシュファイルシステム（usbflash1）とスイッチ番号が表示されることがあります。 （注） **show inventory** コマンドの出力に「usbflash1」と表示されるのは、デバイスが「Disabled and Unlocked」状態または「Enabled and Unlocked」状態の場合のみです。
more `file-url`	SMART エラーおよびドライブの全体的な正常性を示すログを表示します。
show hw-module usbflash1 security status	USB 3.0 SSD 認証ステータスを表示します。

トラブルシューティングのヒント

USB 3.0 SSD の挿入および取り外しのトラブルシューティング

表 *21* : エラーとトラブルシューティング

発生する可能性のあるエラー	トラブルシューティング
挿入後に USB3.0 SSD が検出されない	• Cisco USB 3.0 SSD を使用しているかどうかを確認します。使用していない場合は、デバイスからドライブを取り外し、Cisco USB 3.0 SSD と交換します。 • Cisco USB 3.0 SSD を使用していて、システムがドライブを検出できない場合は、USB 3.0 SSD を取り外して再度挿入します。それでも障害が発生する場合は、USB が不良品である可能性があります。
USB 3.0 SSD の取り外し後にコンソールに表示されるエラーメッセージ： `*Mar 20 00:48:16.353: %IOSXE-4-PLATFORM: Switch 1 R0/0: kernel: xhci_hcd 0000:00:14.0: Cannot set link state.` `*Mar 20 00:48:16.353: %IOSXE-3-PLATFORM: Switch 1 R0/0: kernel: usb usb4-port1: cannot disable (err = -32)` `*May 10 01:12:49.603: %IOSXE-3-PLATFORM: Switch 3 R0/0: kernel: JBD2: Error -5 detected when updating journal superblock for sda1-8.`	**unmount** コマンドを実行した後、デバイスから USB 3.0 SSD を取り外します。詳細については、スイッチまたはスイッチスタックからの USB 3.0 SSD のマウント解除（139 ページ）を参照してください。
シスコ以外の USB 3.0 SSD を挿入すると、コンソールに次のエラーメッセージが表示されます。 `%IOSXEBOOT-4-SSD_MOUNT_LOG: (local/local): ***INFO: Not a CISCO SSD - Cannot be used***`	デバイスから USB を取り外し、Cisco USB 3.0 SSD と交換します。

パスワード認証に関するトラブルシューティング

表 *22 :* エラーとトラブルシューティング

発生する可能性のあるエラー	トラブルシューティング
挿入後に USB3.0 SSD が検出されない	**show hw-module usbflash1 security status** コマンドを実行し、出力の [USB Authentication Status] フィールドを確認します。出力の [USB Authentication Status] フィールドに [Enabled and Locked] が表示されている場合は、次のいずれかを実行します。 • **hw-module switch 1** *switch-number* **usbflash1 security unlock password** *usb-password* コマンドを使用して、ドライブを一時的にロック解除します。 • スイッチで USB 3.0 SSD パスワードを設定します。「スイッチでの USB 3.0 SSD パスワードの設定（140 ページ）」を参照してください。
USB 3.0 SSD のパスワードが、スイッチの実行中のコンフィギュレーションに保存されているパスワードと一致しません。スイッチに次のエラーメッセージが表示されます。 `*Oct 19 19:32:04.094: %IOSD_INFRA-6-IFS_DEVICE_OIR: Device usbflash1 added` `*Oct 19 19:32:04.138: Warning: Configured password on SWITCH does not match with that on DRIVE.` `Please remove password from SWITCH first and then from DRIVE to re-configure.`	次の手順を実行します。 • スイッチからパスワードを削除し、正しいパスワードを使用するようにスイッチを再設定します。「スイッチでの USB 3.0 SSD パスワードの設定（140 ページ）」を参照してください。

発生する可能性のあるエラー	トラブルシューティング
ドライブパスワードが設定されているスイッチにパスワードがない USB 3.0 SSD が挿入されています。スイッチで設定されたパスワードを使用したディスクのロック解除は失敗し、スイッチに次のメッセージが表示されます。 `*Dec 14 00:01:00.374: %IOSD_INFRA-6-IFS_DEVICE_OIR: Device usbflash1 added` `*Dec 14 00:01:00.430: ERROR: No password configured on DRIVE. Remove password from SWITCH to re-configure.`	次の操作を行ってください。 1. ドライブ USB 3.0 SSD でセキュリティを有効にします。「USB 3.0 SSD でのパスワードセキュリティの有効化（139 ページ）」を参照してください。 2. スイッチのパスワードを再設定します。「スイッチでの USB 3.0 SSD パスワードの設定（140 ページ）」を参照してください。
ドライブパスワードが設定されていないスイッチにパスワードが設定された USB 3.0 SSD が挿入されています。ディスクのロック解除は失敗し、スイッチに次のメッセージが表示されます。 `*Oct 19 19:36:18.003: %IOSD_INFRA-6-IFS_DEVICE_OIR: Device usbflash1 added` `*Oct 19 19:36:18.028: Warning: No password configured on SWITCH. Remove password from DRIVE to re-configure`	次のいずれかを実行します。 • ドライブに設定されているパスワードを無効にします。「USB 3.0 SSD でのパスワードセキュリティの無効化（142 ページ）」を参照してください。 • スイッチでパスワードを設定します。「スイッチでの USB 3.0 SSD パスワードの設定（140 ページ）」を参照してください。
`Disabled and locked` 状態の USB 3.0 SSD は、ハードウェアの破損により USB ドライブが使用できなくなったことを示します。	ドライブのロックを解除して有効にするには、TAC にお問い合わせください。

USB 3.0 SSD の設定例

例：USB 3.0 SSD 認証ステータスの表示

この例では、4 つのスイッチを備えたスイッチスタックの USB 3.0 SSD 認証ステータスを示します。

```
# show hw-module usbflash1 security status

Switch#   USB Authentication        Status
-----------------------------------------------------
1         USB Not Present          □ USB 3.0 is not present
2         Disabled and Unlocked    □ Security is disabled & the drive in unlocked state
(Default state if USB is  present)
```

```
3         Enabled and Locked     □ Security Enabled and the drive in locked state
4         Enabled and Unlocked   □ Security Enabled and the drive in unlocked state
```

ドライブが *Enabled and Unlocked* または *Disabled and Unlocked* 状態の場合、ドライブをフォーマットし、読み取り、書き込み、削除、コピーなどの通常のファイルシステム操作を実行できます。

例：ファイルシステムの確認

次に、特権 EXEC モードでの **dir usbflash1:/** コマンドの出力例を示します。

```
Switch#dir usbflash1:

Directory of usbflash1:/
11  drwx            16384   Oct 9 2015 01:49:18 +00:00  lost+found
3145729  drwx             4096   Oct 9 2015 04:10:41 +00:00  test
118014062592 bytes total (111933120512 bytes free)
```

次に、デバイススタックでの **dir usbflash1:***switch_num***:** コマンドの出力例を示します。

```
Switch#dir  usbflash1-2:
Directory of usbflash1-2:/

11 drwx 16384 Jun 8 2018 21:35:39 +00:00 lost+found

118014083072 bytes total (111933390848 bytes free)
```

または、**dir stby-usbflash1:** コマンドを使用して、スタンバイスイッチのファイルシステムにアクセスできます。

```
Switch#dir  stby-usbflash1:
Directory of usbflash1-3:/
11  drwx            16384  May 16 2018 23:32:43 +00:00  lost+found
118014083072 bytes total (110358429696 bytes free)
```

usbflash1 のファイルシステム情報を表示するには、特権 EXEC モードで **show usbflash1: filesystem** コマンドを使用します。

```
Switch#show usbflash1: filesystem
Filesystem: usbflash1
Filesystem Path: /vol/usb1
Filesystem Type: ext4
```

例：物理インベントリ情報の確認

USB 3.0 SSD ハードウェアの物理インベントリ情報を表示するには、**show inventory** コマンドを使用します。

```
Switch#show inventory

NAME: "usbflash1", DESCR: "usbflash1"
PID: SSD-120G          , VID: STP21460FN9, SN: V01
```

次に、デバイススタックの **show inventory** コマンドの出力例を示します。

```
Switch#show inventory

NAME: "usbflash1", DESCR: "usbflash1"
PID: SSD-120G          , VID: STP21460FN9, SN: V01
```

```
NAME: "usbflash1-3", DESCR: "usbflash1-3"
PID: SSD-120G          , VID: STP21310001, SN: V01
```

例：ドライブの正常性の確認

ドライブの全体的な正常性を確認するには、特権 EXEC モードで **more flash:smart_overall_health.log** コマンドを使用します。

```
Switch#more flash:smart_overall_health.log

=== START OF READ SMART DATA SECTION ===
SMART overall-health self-assessment test result: PASSED
```

正常性エラーログを確認するには、特権 EXEC モードで **more crashinfo:tracelogs/smart_errors.log** コマンドを使用します。

```
Switch#more crashinfo:tracelogs/smart_errors.log
%IOSXEBOOT-4-SMART_LOG: (local/local): Mon Jan 4 00:13:10 Universal 2016 INFO: Starting
 SMART daemon
```

（注） システムは、smart_errors.log に警告を表示することがあります。flash/smart_overall_health.log の全体的な正常性のセルフアセスメントに PASSED と表示されている場合は、これらを無視できます。

USB 3.0 SSD 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	パスワード認証付き USB 3.0 SSD	USB 3.0 SSD は、汎用ストレージデバイスおよびアプリケーションホスティングデバイスとして使用するための追加の 120 GB ストレージを提供します。 パスワード認証機能を使用すると、USB 3.0 SSD デバイスにパスワードを設定して、不正アクセスや関連するリスクからドライブを保護できます。

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 17 章

外部 USB Bluetooth ドングルの設定

外部 USB Bluetooth ドングルの設定の制約事項

- Bluetooth バージョン 4.0 のみがサポートされています。
- 外部 USB Bluetooth ドングルは、IPv4 アドレス範囲内で設定されている Cisco Catalyst 9000 シリーズ スイッチでのみサポートされます。
- スタッキングモードでは、外部 USB Bluetooth ドングルをアクティブなスイッチで有効にする必要があります。
- ステートフル スイッチオーバー（SSO）後、外部 USB Bluetooth ドングルを新しいアクティブなスイッチインターフェイスで有効にする必要があります。
- 次の構成では、外部 USB Bluetooth ドングルはサポートされません。
  - Quality of Service（QoS）
  - アクセス コントロール リスト（ACL）

外部 USB Bluetooth ドングルについて

接続された外部 USB Bluetooth ドングルは外部デバイスの Bluetooth ホストとして動作し、スイッチ上の管理ポートとして機能します。外部 USB Bluetooth ドングルは、スマートフォン、ラップトップ、タブレットなどの Bluetooth 対応外部デバイスとペアリングできます。

外部 USB Bluetooth ドングルは、スタンドアロンモードまたはスタッキングモードの両方で設定されたスイッチでサポートされます。

サポートされている外部 USB Bluetooth ドングル

次の外部 USB Bluetooth ドングルがサポートされています。

- BTD-400 Bluetooth 4.0 アダプタ（Kinivo 社製）
- Bluetooth 4.0 USB アダプタ（ASUS 社製）
- ミニ Bluetooth ワイヤレス USB 4.0 ドングルアダプタ（Adnet 社製）
- Bluetooth 4.0 USB アダプタ（Insignia 社製）

スイッチでの外部 USB Bluetooth ドングルの設定方法

スイッチで外部 USB Bluetooth ドングルを設定するには、次の手順を実行します。

手順

ステップ 1 外部 USB Bluetooth ドングルをスイッチの USBタイプ A ポートに接続します。

（注） 外部 USB Bluetooth ドングルは、デバイスの電源を入れる前、またはデバイスの動作中に接続できます。

ステップ 2 スイッチでグローバル コンフィギュレーション モードを開始し、外部 USB Bluetooth ドングルがスイッチに接続されていることを確認します。

```
Device> enable
Device# show platform hardware bluetooth
Controller:0:1a:7d:da:71:13
Type:Primary
Bus:USB
State:DOWN
Name:HCI Version:
```

ステップ 3 インターフェイス コンフィギュレーション モードで **enable** コマンドを使用して Bluetooth インターフェイスを有効にします。

```
Device# configure terminal
Device(config)# interface bluetooth 0/4
Device(config-if)# enable
```

ステップ 4 **no shutdown** コマンドを入力し、デバイスの再起動後に Bluetooth インターフェイスを自動的に再起動します。

```
Device(config-if)# no shutdown
```

ステップ 5 **bluetooth pin** *pin* コマンドを使用してペアリングピンを設定します。

```
Device(config-if)# bluetooth pin 1111
```

または

```
Device(config-if)# exit
Device(config)# bluetooth pin 1111
```

（注） **bluetooth pin** コマンドはグローバル コンフィギュレーション モードで使用することをお勧めします。

ステップ 6 外部デバイスの Bluetooth 設定をオンにします。外部デバイスで、ホスト名に基づいて Bluetooth 対応スイッチを選択します。

ステップ 7 外部デバイスがインターネットに接続できるようにするには、外部デバイスのネットワーク設定を有効にします。

スイッチでの Bluetooth 設定の確認

Bluetooth 設定をモニタリングするには、特権 EXEC モードで次のコマンドを使用します。

表 *23* : デバイスでの *Bluetooth* 設定をモニタするコマンド

コマンド	目的
show ip interface bluetooth 0/4	Bluetooth インターフェイスのユーザビリティ ステータスを表示します。
show platform hardware bluetooth	Bluetooth インターフェイスに関する情報を表示します。
show running | include pin	現在の Bluetooth ピンを表示します。

外部 Bluetooth ドングルの設定の機能履歴

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	外部 Bluetooth ドングルの設定	外部 USB Bluetooth ドングルは外部デバイスの Bluetooth ホストとして動作し、スイッチの管理ポートとして機能します。

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。