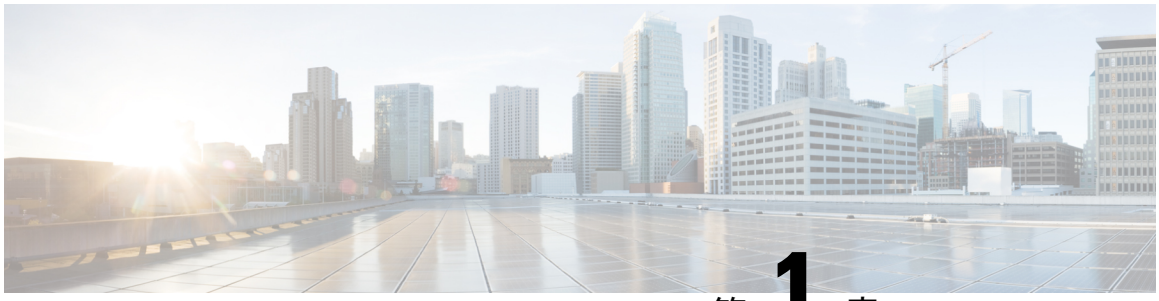
Cisco IOS リリース 15.2(7)E（Catalyst 1000 スイッチ）ネットワーク管理コンフィギュレーション ガイド

初版：2019 年 12 月 25 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

Cisco IOS Configuration Engine の設定

Configuration Engine を設定するための前提条件

- ユーザが接続している Configuration Engine インスタンスの名前を取得します。
- CNS は、イベントバスとコンフィギュレーション サーバの両方を使用してデバイスに設定を提供するので、設定済みのデバイスごとに ConfigID と DeviceID の両方を定義する必要があります。
- **cns config partial** 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設定されたすべてのデバイスは、イベントバスにアクセスする必要があります。したがって、（デバイスを起源とする）DeviceID が、Cisco Configuration Engine 内の対応するデバイス定義の DeviceID と一致する必要があります。ユーザが接続しているイベント バスのホスト名を把握する必要があります。

Configuration Engine の設定に関する制約事項

- コンフィギュレーション サーバの 1 つのインスタンスでは、設定済みの 2 つのデバイスが同じ ConfigID 値を共有できません。
- イベントバスの 1 つのインスタンスでは、設定済みの 2 つのデバイスが同じ DeviceID 値を共有できません。

Configuration Engine の設定について

ここでは、Configuration Engine の設定方法について説明します。

Cisco Configuration Engine ソフトウェア

Cisco Configuration Engine は、ネットワーク管理ユーティリティ ソフトウェアで、ネットワーク デバイスおよびサービスの配置と管理を自動化するためのコンフィギュレーション サービスとして機能します。各 Cisco Configuration Engine は、シスコデバイス（スイッチとルータ）のグループとデバイスが提供するサービスを管理し設定を保存して、必要に応じて配信します。Cisco Configuration Engine は、デバイス固有のコンフィギュレーション変更を生成してデバイスに送信し、コンフィギュレーション変更を実行して結果をログに記録することにより、初期設定とコンフィギュレーションの更新を自動化します。

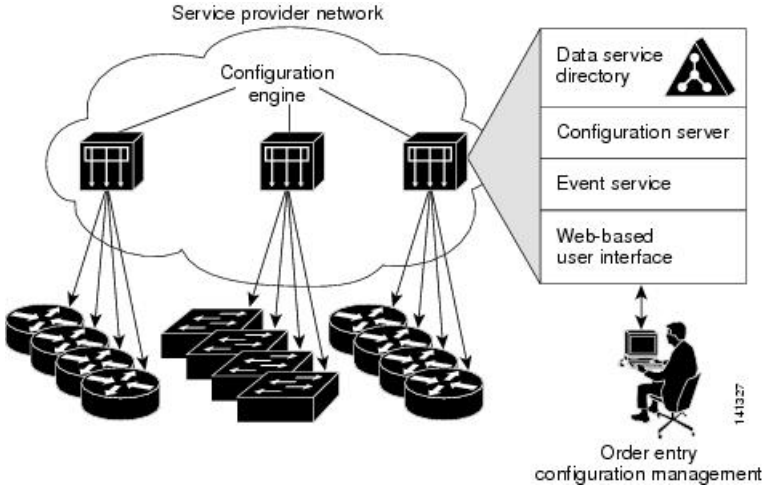
Cisco Configuration Engine は、スタンドアロン モードとサーバ モードをサポートし、次の Cisco Networking Service（CNS）コンポーネントがあります。

- コンフィギュレーション サービス
  - Web サーバ
  - ファイル マネージャ
  - ネームスペース マッピング サーバ
- イベント サービス（イベント ゲートウェイ）
- データ サービス ディレクトリ（データ モデルおよびスキーマ）

（注） Cisco Configuration Engine のサポートは、今後のリリースで廃止されます。『Cisco Plug and Play Feature Guide』に説明されている構成を使用してください。

スタンドアロン モードでは、内部に組み込まれたディレクトリ サービスがサポートされます。このモードでは、外部ディレクトリまたはその他のデータ ストアは必要ありません。サーバ モードでは、ユーザが定義した外部ディレクトリの使用がサポートされます。

図 *1 : Cisco Configuration Engine* のアーキテクチャの概要



コンフィギュレーション サービス

コンフィギュレーション サービスは、Cisco Configuration Engine の中核コンポーネントです。デバイス上にある Cisco IOS CNS エージェントと連携して動作するコンフィギュレーション サーバで構成されています。コンフィギュレーション サービスは、初期設定と論理グループによる大規模な再設定のために、デバイスとサービスの設定をデバイスに配信します。スイッチはネットワーク上で初めて起動する際に、コンフィギュレーション サービスから初期設定を受信します。

コンフィギュレーション サービスは CNS イベント サービスを使用して設定変更イベントを送受信し、成功および失敗の通知を送信します。

コンフィギュレーション サーバは Web サーバであり、コンフィギュレーション テンプレートと組み込み型ディレクトリ（スタンドアロン モード）またはリモート ディレクトリ（サーバ モード）に保存されているデバイス固有の設定情報を使用します。

コンフィギュレーションテンプレートは、CLI（コマンドライン インターフェイス）コマンド形式で静的な設定情報を含んだテキスト ファイルです。テンプレートでは、変数は、Lightweight Directory Access Protocol（LDAP）URL を使用して指定します。この URL はディレクトリに保存されているデバイス固有の設定情報を参照します。

Cisco IOS エージェントは受信したコンフィギュレーション ファイルの構文をチェックし、イベントを発行して構文チェックが成功または失敗したかを表示します。コンフィギュレーション エージェントは設定をただちに適用することも、あるいは同期化イベントをコンフィギュレーション サーバから受信するまで適用を遅らせることもできます。

イベント サービス

Cisco Configuration Engine は、設定イベントの受信および生成にイベント サービスを使用します。イベント サービスはイベント エージェント、イベント ゲートウェイから構成されます。

イベントエージェントはデバイス上にあり、デバイスと Cisco Configuration Engine のイベントゲートウェイ間の通信を容易にします。

イベント サービスは、非常に有効なパブリッシュ サブスクライブ通信方式です。イベント サービスは、サブジェクトベースのアドレス指定を使用して、メッセージを宛先に送信します。サブジェクトベースのアドレス表記法では、メッセージおよび宛先には簡単で均一なネームスペースを定義します。

名前空間マッパー

Cisco Configuration Engine はネームスペース マッパー（NSM）を備えています。これは、アプリケーション、デバイスまたはグループ ID、およびイベントに基づいてデバイスの論理グループを管理するための検索サービスを提供します。

Cisco IOS デバイスは、たとえば cisco.cns.config.load といった、Cisco IOS ソフトウェアで設定されたサブジェクト名と一致するイベント サブジェクト名のみを認識します。ネームスペース マッピング サービスを使用すると、希望する命名規則を使用することでイベントを指定できます。サブジェクト名でデータ ストアにデータを入力した場合、NSM はイベント サブジェクト名ストリングを、Cisco IOS が認識するものに変更します。

サブスクライバの場合、一意のデバイス ID とイベントが指定されると、ネームスペース マッピング サービスは、サブスクライブ対象のイベント セットを返します。同様にパブリッシャの場合、一意のグループ ID、デバイス ID、およびイベントが指定されると、マッピング サービスは、パブリッシュ対象のイベント セットを返します。

Cisco Networking Service ID およびデバイスのホスト名

Cisco Configuration Engine は、設定対象の各デバイスに一意の識別子が関連付けられていることを前提としています。一意の識別子は複数の同義語を持つことができますが、各同義語は特定のネームスペース内で一意です。イベント サービスは、ネームスペースの内容を使用してメッセージのサブジェクトベース アドレス指定を行います。

Cisco Configuration Engine は、イベント バス用とコンフィギュレーション サーバ用の 2 つの名前空間を交差します。コンフィギュレーション サーバのネームスペースでは、*ConfigID* という用語がデバイスの一意な識別子です。イベント バスのネームスペースでは、*DeviceID* という用語がデバイスの CNS 一意識別子です。

ConfigID

設定対象のデバイスはそれぞれ一意の ConfigID を持ちます。これは Cisco Configuration Engine ディレクトリからデバイス CLI 属性の対応するセットを取得するためのキーとなります。デバイスで定義された ConfigID は、Cisco Configuration Engine 上の対応するデバイス定義の ConfigID と一致する必要があります。

ConfigID は起動時に固定され、デバイスホスト名を再設定した場合でもデバイスを再起動するまで変更できません。

DeviceID

イベント バスに参加している設定対象デバイスはそれぞれ一意の DeviceID を持ちます。これはデバイス ソース アドレスに似ており、これによってデバイスをバス上の特定の宛先として指定できます。

DeviceID の起源は、デバイスの Cisco IOS ホスト名によって定義されます。ただし、DeviceID 変数はデバイスに隣接するイベント ゲートウェイ内に存在し、そこで使用されます。

イベント バス上の論理的な Cisco IOS 終端地点はイベント ゲートウェイに組み込まれており、イベント ゲートウェイがデバイスの代わりにプロキシの役割を果たします。イベント ゲートウェイは、イベント バスに対してデバイスとデバイスに対応する DeviceID を表します。

デバイスは、イベント ゲートウェイへの接続が成功すると、ただちに自身のホスト名をイベント ゲートウェイに宣言します。接続が確立されるたびに、イベント ゲートウェイは DeviceID 値を Cisco IOS ホスト名に組み合わせます。イベントゲートウェイは、デバイスと接続している間、この DeviceID 値を保持します。

ホスト名および DeviceID

DeviceID はイベント ゲートウェイに接続した時点で固定され、デバイスのホスト名が再設定された場合にも変更されません。

デバイスでデバイスホスト名を変更するとき、DeviceID を更新する唯一の方法は、デバイスとイベントゲートウェイ間の接続を切断することです。DeviceID 更新の手順については、以下の「関連項目」を参照してください。

接続が再確立されると、デバイスは変更したホスト名をイベントゲートウェイに送信します。イベント ゲートウェイは DeviceID を新しい値に再定義します。

⚠

注意 Cisco Configuration Engine ユーザインターフェイスを使用するときは、最初に DeviceID フィールドを、デバイスが前ではなく後に取得するホスト名値に設定する必要があります。Cisco IOS CNS エージェント用に設定を再初期化する必要があります。そのようにしないと、後続の部分的なコンフィギュレーション コマンド操作で誤動作が発生する可能性があります。

ホスト名、DeviceID、および ConfigID

スタンドアロン モードでは、デバイスのホスト名の値が設定されている場合、コンフィギュレーション サーバからイベントがホスト名に送信されるときに、設定されたホスト名が DeviceID として使用されます。ホスト名が設定されていない場合、イベントはデバイスの cn=<*value*>で送信されます。

サーバ モードでは、ホスト名は使用されません。このモードでは、バス上のイベント送信には常に一意の DeviceID 属性が使用されます。この属性が設定されていない場合はデバイスを更新できません。

Cisco Configuration Engine で **Setup** を実行する場合、これらの属性および関連する属性（タグ値のペア）を設定します。

自動 CNS 設定

デバイスの自動 CNS 設定をイネーブルにするには、まずこのトピックに示す前提条件を完了する必要があります。条件設定を完了したらデバイスの電源を入れます。**setup** プロンプトでは何も入力しません。デバイスが初期設定を開始します。コンフィギュレーション ファイル全体がデバイスにロードされると作業は完了です。

初期設定中の動作については、「関連項目」を参照してください。

表 *1*: 自動設定イネーブル化の条件

デバイス	必要な設定
アクセス デバイス	出荷時の設定（コンフィギュレーション ファイルなし）
ディストリビューション デバイス	• IP ヘルパー アドレス • DHCP リレー エージェントをイネーブルにする[1] • IP ルーティング（デフォルト ゲートウェイとして使用する場合）
DHCP サーバ	• IP アドレスの割り当て • TFTP サーバの IP アドレス • TFTP サーバのブートストラップ コンフィギュレーション ファイルへのパス • デフォルト ゲートウェイの IP アドレス
TFTP サーバ	• デバイスと Configuration Engine との通信を可能にする CNS コンフィギュレーション コマンドを含むブートストラップ コンフィギュレーション ファイル • （デフォルトのホスト名の代わりに）デバイス MAC アドレスまたはシリアル番号のいずれかを使用して ConfigID および EventID を生成するように設定されたデバイス • デバイスにコンフィギュレーション ファイルをプッシュするように設定された CNS イベントエージェント
CNS Configuration Engine	デバイス タイプ別の 1 つまたは複数のテンプレートで、テンプレートにデバイスの ConfigID がマッピングされています。

[1] DHCP リレーは、DHCP サーバがクライアントとは異なるサブネット上にある場合にのみ必要です。

Configuration Engine の設定方法

ここでは、Configuration Engine の設定方法について説明します。

CNS イベント エージェントのイネーブル化

（注） デバイス上で CNS イベントエージェントをイネーブルにしてから、CNS 設定エージェントをイネーブルにする必要があります。

デバイス上で CNS イベントエージェントをイネーブルにするには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**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	**cns event** {*hostname* | *ip-address*} [*port-number*] [[**keepalive** *seconds retry-count*] [**failover-time** *seconds*] [**reconnect-time** *time*] | **backup**] 例： `Device(config)# cns event 10.180.1.27 keepalive 120 10`	イベント エージェントをイネーブルにして、ゲートウェイ パラメータを入力します。 • {*hostname* | *ip-address*} に、イベント ゲートウェイのホスト名または IP アドレスを入力します。 • （任意）*port number* に、イベント ゲートウェイのポート番号を入力します。デフォルトのポート番号は 11011 です。 • （任意）**keepalive** *seconds* に、デバイスがキープアライブメッセージを送信する間隔を入力します。*retry-count* に、キープアライブメッセージへの応答がない場合に接続を終了するまでのデバイスのメッセー

	コマンドまたはアクション	目的
		ジ送信回数を入力します。デフォルト値はいずれも 0 です。 • （任意）**failover-time** *seconds* に、バックアップゲートウェイが確立された後にデバイスがプライマリ ゲートウェイ ルートを待つ時間を入力します。 • （任意）**reconnect-time** *time* に、デバイスがイベントゲートウェイに再接続しようとする前の最大時間間隔を入力します。 • （任意）バックアップゲートウェイであることを示す場合は、**backup** を入力します（省略した場合は、プライマリ ゲートウェイになります）。 （注） **encrypt** および **clock-timeout** *time* キーワードは、コマンドラインのヘルプストリングに表示されますが、サポートされていません。
ステップ 4	**end** 例： `Device (config)#end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

次のタスク

イベントエージェントに関する情報を確認するには、**show cns event connections** コマンドを特権 EXEC モードで使用します。

CNS イベントエージェントをディセーブルにするには、**no cns event** { *ip-address* | *hostname* } 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。

DeviceID の更新

デバイス上でホスト名を変更するときに DeviceID を更新するには、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	**show cns config connections** 例： `Device# show cns config connections`	CNS イベント エージェントがゲートウェイに接続しているか、接続されているか、またはアクティブか、およびイベントエージェントに使用されているゲートウェイ、そのIPアドレス、およびポート番号を表示します。
ステップ 3	CNSイベントエージェントがイベントゲートウェイに正しく接続されていることを確認します。	次の点について、**show cns config connections** の出力調べます。 • 接続がアクティブになっている。 • 接続で現在設定されているデバイスホスト名を使用している。DeviceID はこれらの手順を使用して、新しいホスト名の設定に対応するように更新されます。
ステップ 4	**show cns event connections** 例： `Device# show cns event connections`	デバイスのイベント接続情報を表示します。
ステップ 5	ステップ 4 の出力に基づいて、次に示す現在接続されている接続に関する情報を記録します。この手順の以降のス	

	コマンドまたはアクション	目的
	テップでIPアドレスとポート番号を使用します。	
ステップ 6	**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 7	**no cns event** *ip-address port-number* 例： `Device(config)# no cns event 172.28.129.22 2012`	このコマンドで、ステップ 5 で記録したIPアドレスとポート番号を指定します。 このコマンドで、デバイスとイベントゲートウェイ間の接続が解除されます。最初に接続を解除し、次にこの接続を再確立して、DeviceID を更新する必要があります。
ステップ 8	**cns event** *ip-address port-number* 例： `Device(config)# cns event 172.28.129.22 2012`	このコマンドで、ステップ 5 で記録したIPアドレスとポート番号を指定します。 このコマンドで、デバイスとイベントゲートウェイ間の接続が再確立されます。
ステップ 9	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 10	**show cns event connections** からの出力を調べて、デバイスとイベント接続間の接続が再確立されていることを確認します。	
ステップ 11	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 12	**copy running-config startup-config** 例： `Device# copy running-config`	（任意）コンフィギュレーションファイルに設定を保存します。

	コマンドまたはアクション	目的
	startup-config	

CNS 設定のモニタリング

表 *2 : CNS show* コマンド

コマンド	目的
show cns config connections `Device# show cns config connections`	CNS Cisco IOS CNS エージェントの接続のステータスを表示します。
show cns config outstanding `Device# show cns config outstanding`	開始されたがまだ終了していない差分（部分）CNS 設定に関する情報を表示します。
show cns config stats `Device# show cns config stats`	Cisco IOS CNS エージェントに関する統計情報を表示します。
show cns event connections `Device# show cns event connections`	CNS イベント エージェントの接続のステータスを表示します。
show cns event gateway `Device# show cns event gateway`	デバイスのイベントゲートウェイ情報を表示します。
show cns event stats `Device# show cns event stats`	CNS イベント エージェントに関する統計情報を表示します。
show cns event subject `Device# show cns event subject`	アプリケーションによってサブスクライブされたイベント エージェントのサブジェクト一覧を表示します。

その他の参考資料

関連資料

関連項目	マニュアル タイトル
Configuration Engine のセットアップ	『*Cisco Configuration Engine Installation and Setup Guide, 1.5 for Linux*』 https://www.cisco.com/en/US/docs/net_mgmt/configuration_engine/1.5/installation_linux/guide/setup_1.html

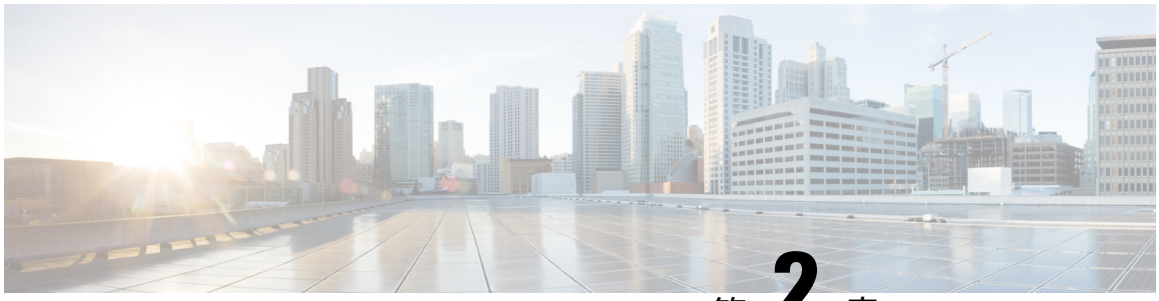
Cisco IOS Configuration Engine の機能情報

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 *3 : Cisco IOS Configuration Engine* の機能情報

機能名	リリース	機能情報
Cisco IOS Configuration Engine	Cisco IOS Release 15.2(7)E1	この機能が導入されました。



第 2 章

Cisco Discovery Protocol の設定

Cisco Discovery Protocol は、シスコデバイス上で動作し、ネットワーキング アプリケーションが直接接続された付近のデバイスに関して学習できるようにする、メディア独立型かつネットワーク独立型のレイヤ2プロトコルです。このプロトコルによってシスコデバイスが検出されてその設定状態が特定され、異なるネットワーク層プロトコルを使用するシステムが相互に学習できるようになることで、デバイスの管理が容易になります。

このモジュールでは、Cisco Discovery Protocol バージョン 2 とその SNMP での動作について説明します。

Cisco Discovery Protocol について

ここでは、Cisco Discovery Protocol について説明します。

Cisco Discovery Protocol の概要

Cisco Discovery Protocol は、すべてのシスコデバイス（ルータ、ブリッジ、アクセスサーバ、コントローラ、およびスイッチ）のレイヤ 2（データリンク層）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、既知のデバイスのネイバーであるシスコデバイスを検出することができます。また、下位レイヤのトランスペアレント プロトコルが稼働しているネイバーデバイスのデバイスタイプや、SNMP エージェントアドレスを学習することもできます。この機能によって、アプリケーションからネイバー デバイスに SNMP クエリーを送信できます。

Cisco Discovery Protocol は、サブネットワーク アクセス プロトコル（SNAP）をサポートしているすべてのメディアで動作します。Cisco Discovery Protocol はデータリンク層でのみ動作するため、異なるネットワーク層プロトコルをサポートする2つのシステムで互いの情報を学習できます。

Cisco Discovery Protocol が設定された各デバイスはマルチキャストアドレスに定期的にメッセージを送信して、SNMP メッセージを受信可能なアドレスを 1 つまたは複数アドバタイズします。このアドバタイズには、受信側デバイスで Cisco Discovery Protocol 情報を廃棄せずに保持する時間を表す存続可能時間、つまりホールドタイム情報も含まれます。各デバイスは他のデバイスから送信されたメッセージも待ち受けて、ネイバー デバイスについて学習します。

Cisco Discovery Protocol はデバイス上で Network Assistant をイネーブルにすることで、ネットワークをグラフィカルに表示できます。デバイスは Cisco Discovery Protocol を使用してクラスタ候補を検出し、クラスタメンバ、およびコマンドデバイスから最大3台（デフォルト）離れたクラスタ対応の他のデバイスについての情報を維持します。

Cisco Discovery Protocol のデフォルト設定

次の表に、Cisco Discovery Protocol のデフォルト設定を示します。

機能	デフォルト設定
Cisco Discovery Protocol グローバル状態	イネーブル
Cisco Discovery Protocol インターフェイス状態	イネーブル
Cisco Discovery Protocol タイマー（パケット更新頻度）	60 秒
Cisco Discovery Protocol 保持時間（廃棄前）	180 秒
Cisco Discovery Protocol バージョン 2 アドバタイズメント	イネーブル

Cisco Discovery Protocol の設定方法

ここでは、Cisco Discovery Protocol の設定方法について説明します。

Cisco Discovery Protocol の特性の設定

次の Cisco Discovery Protocol の特性を設定できます。

- Cisco Discovery Protocol アップデートの頻度
- 破棄するまで情報を保持する時間の長さ
- バージョン 2 アドバタイズメントを送信するかどうか

（注） ステップ 3 ～ 5 はすべて任意であり、どの順番で実行してもかまいません。

次の手順に従って、Cisco Discovery Protocol の特性を設定します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • 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	**cdp timer** *seconds* 例： `Device(config)# cdp timer 20`	（任意）Cisco Discovery Protocol 更新の送信頻度を秒単位で設定します。 指定できる範囲は 5 ～ 254 です。デフォルトは 60 秒です。
ステップ 4	**cdp holdtime** *seconds* 例： `Device(config)# cdp holdtime 60`	（任意）受信デバイスがこのデバイスから送信された情報を破棄せずに保持する時間を指定します。 指定できる範囲は 10 ～ 255 秒です。デフォルトは 180 秒です。
ステップ 5	**cdp advertise-v2** 例： `Device(config)# cdp advertise-v2`	（任意）バージョン 2 アドバタイズを送信するように Cisco Discovery Protocol を設定します。 これは、デフォルトの状態です。
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 8	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

次のタスク

デフォルト設定に戻すには、Cisco Discovery Protocol コマンドの **no** 形式を使用します。

Cisco Discovery Protocol のディセーブル化

Cisco Discovery Protocol はデフォルトでイネーブルになっています。

（注） デバイスクラスタと他のシスコデバイス（Cisco IP Phone など）は、Cisco Discovery Protocol メッセージを定期的に交換します。Cisco Discovery Protocol をディセーブルにすると、クラスタ検出が中断され、デバイスの接続が切断されます。

Cisco Discovery Protocol デバイス検出機能をディセーブルにするには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • 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	**no cdp run** 例： `Device(config)# no cdp run`	Cisco Discovery Protocol を無効にします。
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

次のタスク

Cisco Discovery Protocol を使用するには、再度有効にする必要があります。

Cisco Discovery Protocol の有効化

Cisco Discovery Protocol はデフォルトでイネーブルになっています。

（注） デバイスクラスタと他のシスコデバイス（Cisco IP Phone など）は、Cisco Discovery Protocol メッセージを定期的に交換します。Cisco Discovery Protocol をディセーブルにすると、クラスタ検出が中断され、デバイスの接続が切断されます。

ディセーブルになっている Cisco Discovery Protocol をイネーブルにするには、次の手順を実行します。

始める前に

Cisco Discovery Protocol がディセーブルになっていないと、イネーブルにはできません。

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	**cdp run** 例： `Device(config)# cdp run`	Cisco Discovery Protocol がディセーブルになっている場合にイネーブルにします。
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

次のタスク

Cisco Discovery Protocol がイネーブルになっていることを表示するには、 **show run all** コマンドを使用します。**show run** を入力しただけでは、Cisco Discovery Protocol がイネーブルになっていることが表示されない場合があります。

インターフェイス上で Cisco Discovery Protocol をディセーブルにします。

Cisco Discovery Protocol は、Cisco Discovery Protocol 情報を送受信するために、サポートされているすべてのインターフェイスでデフォルトで有効になっています。

（注） デバイスクラスタと他のシスコデバイス（Cisco IP Phone など）は、Cisco Discovery Protocol メッセージを定期的に交換します。Cisco Discovery Protocol をディセーブルにすると、クラスタ検出が中断され、デバイスの接続が切断されます。

（注） Discovery Protocol バイパスはサポートされていないため、ポートが err-disabled ステートになる場合があります。

ポートで Cisco Discovery Protocol をディセーブルにするには、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	**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 1/0/1`	Cisco Discovery Protocol をディセーブルにするインターフェイスを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**no cdp enable** 例： `Device(config-if)# no cdp enable`	ステップ 3 で指定したインターフェイス上で Cisco Discovery Protocol をディセーブルにします。

	コマンドまたはアクション	目的
ステップ 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。

インターフェイス上での Cisco Discovery Protocol のイネーブル化

Cisco Discovery Protocol は、Cisco Discovery Protocol 情報を送受信するために、サポートされているすべてのインターフェイスでデフォルトでイネーブルになっています。このタスクは必須ではありません。

（注） デバイスクラスタと他のシスコデバイス（Cisco IP Phone など）は、Cisco Discovery Protocol メッセージを定期的に交換します。Cisco Discovery Protocol をディセーブルにすると、クラスタ検出が中断され、デバイスの接続が切断されます。

（注） Discovery Protocol バイパスはサポートされていないため、ポートが err-disabled ステートになる場合があります。

ポートでディセーブルになっている Cisco Discovery Protocol をイネーブルにするには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例：	グローバル 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	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	Cisco Discovery Protocol をイネーブルにするインターフェイスを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**cdp enable** 例： `Device(config-if)# cdp enable`	ディセーブルになっているインターフェイスで Cisco Discovery Protocol をイネーブルにします。
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。

Cisco Discovery Protocol のモニタリングとメンテナンス

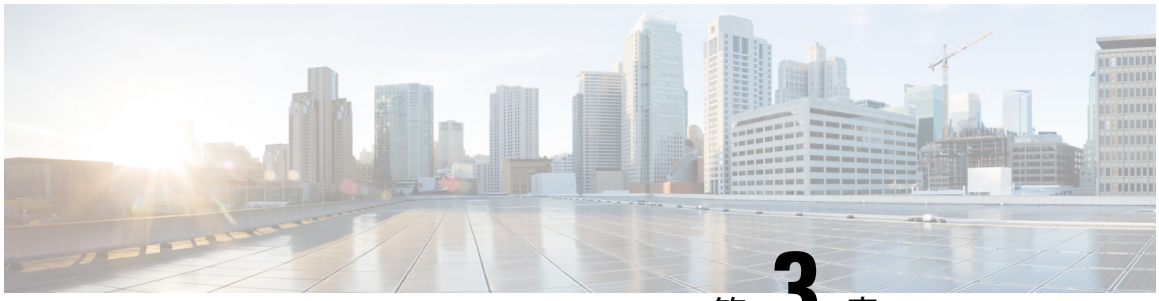
表 *4 : Cisco Discovery Protocol* 情報を表示するためのコマンド

コマンド	説明
clear cdp counters	トラフィック カウンタを 0 にリセットします。
clear cdp table	ネイバーに関する情報の Cisco Discovery Protocol テーブルを削除します。
show cdp	送信間隔、送信したパケットの保持時間などのグローバル情報を表示します。

コマンド	説明
show cdp entry *entry-name* ［**version**］［**protocol**］	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（*）を入力して、すべてのCisco Discovery Protocol ネイバーを表示することも、情報が必要なネイバーの名前を入力することもできます。 また、指定されたネイバー上でイネーブルになっているプロトコルの情報や、デバイス上で稼働しているソフトウェアのバージョン情報が表示されるように、表示内容を制限することもできます。
show cdp interface［*interface-id*］	Cisco Discovery Protocol がイネーブル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。 必要なインターフェイスの情報だけを表示できます。
show cdp neighbors［*interface-id*］［*detail*］	装置タイプ、インターフェイス タイプ、インターフェイス番号、保持時間の設定値、機能、プラットフォーム、ポート ID を含めたネイバー情報を表示します。 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 cdp traffic	Cisco Discovery Protocol カウンタ（送信済み/受信済みパケット数とチェックサム エラー数を含む）を表示します。

Cisco Discovery Protocol の機能の履歴と情報

リリース	変更内容
Cisco IOS Release 15.2(7)E1	この機能が導入されました



第 3 章

簡易ネットワーク管理プロトコルの設定

SNMP の前提条件

サポートされている SNMP バージョン

このソフトウェア リリースは、次の SNMP バージョンをサポートしています。

- SNMPv1：RFC1157 に規定された SNMP（完全インターネット標準）。
- SNMPv2C は、SNMPv2Classic のバルク検索機能を残し、エラー処理を改善したうえで、SNMPv2Classic のパーティ ベースの管理およびセキュリティ フレームワークをコミュニティ ストリング ベースの管理フレームワークに置き換えたものです。次の機能があります。
  - SNMPv2：RFC 1902 ～ 1907 に規定された SNMP バージョン 2（ドラフト版インターネット標準）
  - SNMPv2C：RFC 1901 に規定された SNMPv2 のコミュニティ ストリング ベースの管理フレームワーク（試験版インターネット プロトコル）
- SNMPv3：SNMP のバージョン 3 は、RFC 2273 ～ 2275 に規定されている相互運用可能な標準ベース プロトコルです。SNMPv3 は、ネットワーク上のパケットを認証、暗号化することでデバイスへのアクセスに対するセキュリティを提供します。SNMPv3 は、次のセキュリティ機能を備えています。
  - メッセージの完全性：パケットが伝送中に改ざんされないようにします。

- 認証：有効な送信元からのメッセージであるかどうかを判別します。
- 暗号化：パッケージの内容をミキシングし、許可されていない送信元に内容が読まれることを防止します。

（注） 暗号化を選択するには、**priv** キーワードを入力します。

SNMPv1 と SNMPv2C は、ともにコミュニティベース形式のセキュリティを使用します。エージェントの MIB にアクセスできるマネージャのコミュニティが、IP アドレス アクセス コントロール リストおよびパスワードによって定義されます。

SNMPv2C にはバルク検索機能が組み込まれ、より詳細なエラー メッセージを管理ステーションに報告します。バルク検索機能は、テーブルや大量の情報を検索し、必要な往復回数を削減します。SNMPv2C ではエラー処理機能が改善され、さまざまなエラーを区別するための拡張エラー コードが使用されています。これらのエラーは、SNMPv1 では単一のエラー コードで報告されます。SNMPv2 では、エラー リターン コードでエラー タイプが報告されるようになりました。

SNMPv3 では、セキュリティ モデルとセキュリティ レベルの両方が提供されています。セキュリティ モデルは、ユーザとユーザが属しているグループ用に設定された認証方式です。セキュリティ レベルとは、セキュリティ モデル内で許可されるセキュリティのレベルです。セキュリティ レベルとセキュリティ モデルの組み合わせにより、SNMP パケットを扱うときに使用するセキュリティ方式が決まります。使用可能なセキュリティ モデルは、SNMPv1、SNMPv2C、および SNMPv3 です。

次の表では、この特性を識別し、セキュリティ モデルとセキュリティ レベルの異なる組み合わせを比較します。

表 *5: SNMP* セキュリティ モデルおよびセキュリティ レベル

モデル	レベル	認証	暗号化	結果
SNMPv1	noAuthNoPriv	コミュニティ ストリング	未対応	コミュニティ ストリングの照合を使用して認証します。
SNMPv2C	noAuthNoPriv	コミュニティ ストリング	未対応	コミュニティ ストリングの照合を使用して認証します。
SNMPv3	noAuthNoPriv	ユーザ名	未対応	ユーザ名の照合を使用して認証します。

モデル	レベル	認証	暗号化	結果
SNMPv3	authNoPriv	Message Digest 5（MD5）または Secure Hash Algorithm（SHA）	未対応	HMAC-MD5 アルゴリズムまたは HMAC-SHA アルゴリズムに基づいて認証します。
SNMPv3	authPriv	MD5 または SHA	データ暗号規格（DES）または Advanced Encryption Standard（AES）	HMAC-MD5 アルゴリズムまたは HMAC-SHA アルゴリズムに基づいて認証します。 次の暗号化アルゴリズムで、User-based Security Model（USM）を指定できます。 • CBC-DES（DES-56）規格に基づく認証に加えた DES 56 ビット暗号化 • 3DES 168 ビット暗号化 • AES 128 ビット暗号化、192 ビット暗号化、または 256 ビット暗号化

管理ステーションでサポートされている SNMP バージョンを使用するには、SNMP エージェントを設定する必要があります。エージェントは複数のマネージャと通信できるため、SNMPv1、SNMPv2C、および SNMPv3 を使用する通信をサポートするようにソフトウェアを設定できます。

SNMP の制約事項

バージョンの制約事項

- SNMPv1 は informs をサポートしていません。

SNMP に関する情報

ここでは、簡易ネットワーク管理プロトコルについて説明します。

SNMP の概要

SNMP は、マネージャとエージェント間の通信のメッセージ フォーマットを提供するアプリケーションレイヤプロトコルです。SNMP システムは、SNMP マネージャ、SNMP エージェント、および管理情報ベース（MIB）で構成されます。SNMP マネージャは、Cisco Prime Infrastructure などのネットワーク管理システム（NMS）に統合できます。エージェントと MIB はデバイス上に存在します。デバイスに SNMP を設定するには、マネージャとエージェントの間の関係を定義します。

SNMP エージェントは MIB 変数を格納し、SNMP マネージャはこの変数の値を要求または変更できます。マネージャはエージェントから値を取得したり、エージェントに値を格納したりできます。エージェントは、デバイス パラメータやネットワーク データの保存場所である MIB から値を収集します。また、エージェントはマネージャのデータ取得またはデータ設定の要求に応答できます。

エージェントは非送信請求トラップをマネージャに送信できます。トラップは、ネットワーク上のある状態を SNMP マネージャに通知するメッセージです。トラップは不正なユーザ認証、再起動、リンク ステータス（アップまたはダウン）、MAC アドレス追跡、TCP 接続の終了、ネイバーとの接続の切断などの重要なイベントの発生を意味する場合があります。

SNMP マネージャ機能

SNMP マネージャは、MIB 情報を使用して、次の表に示す動作を実行します。

表 *6 : SNMP* の動作

動作	説明
get-request	特定の変数から値を取得します。
get-next-request	テーブル内の変数から値を取得します。[2]
get-bulk-request[3]	テーブルの複数の行など、通常はサイズの小さい多数のデータ ブロックに分割して送信する必要がある巨大なデータ ブロックを取得します。
get-response	NMS から送信される get-request、get-next-request、および set-request に対して応答します。
set-request	特定の変数に値を格納します。
trap	SNMP エージェントから SNMP マネージャに送られる、イベントの発生を伝える非送信請求メッセージです。

[2] この動作では、SNMP マネージャに正確な変数名を認識させる必要はありません。テーブル内を順に検索して、必要な変数を検出します。
[3] get-bulk コマンドを使用できるのは、SNMPv2 以上に限られます。

SNMP エージェント機能

SNMP エージェントは、次のようにして SNMP マネージャ要求に応答します。

- MIB 変数の取得：SNMP エージェントは NMS からの要求に応答して、この機能を開始します。エージェントは要求された MIB 変数の値を取得し、この値を使用して NMS に応答します。
- MIB 変数の設定：SNMP エージェントは NMS からのメッセージに応答して、この機能を開始します。SNMP エージェントは、MIB 変数の値を NMS から要求された値に変更します。

エージェントで重要なイベントが発生したことを NMS に通知するために、SNMP エージェントは非送信請求トラップ メッセージも送信します。トラップ条件の例には、ポートまたはモジュールがアップまたはダウン状態になった場合、スパニングツリー トポロジが変更された場合、認証に失敗した場合などがあります。

SNMP コミュニティ ストリング

SNMP コミュニティ ストリングは、MIB オブジェクトへのアクセスを認証し、組み込みパスワードとして機能します。NMS がデバイスにアクセスするには、NMS 上のコミュニティストリング定義がデバイス上の 3 つのコミュニティストリング定義の少なくとも 1 つと一致しなければなりません。

コミュニティ ストリングの属性は、次のいずれかです。

- 読み取り専用（RO）：コミュニティ ストリングを除き MIB 内のすべてのオブジェクトに、許可された管理ステーションに対する読み取りアクセス権を与えますが、書き込みアクセスは許可しません。
- 読み取り-書き込み（RW）：MIB 内のすべてのオブジェクトに、許可された管理ステーションに対する読み取りおよび書き込みアクセス権を与えますが、コミュニティストリングへのアクセスは許可しません。
- クラスタを作成すると、コマンドデバイスがメンバデバイスと SNMP アプリケーション間のメッセージ交換を管理します。Network Assistant ソフトウェアは、コマンドデバイス上で最初に設定された RW および RO コミュニティストリングにメンバデバイス番号（@esN、N はデバイス番号）を追加し、これらのストリングをメンバデバイスに伝播します。

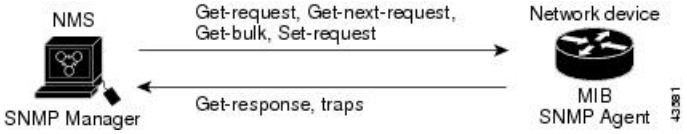
SNMP MIB 変数アクセス

MIB 変数を使用してデバイス変数を設定し、ネットワーク上のデバイスをポーリングして特定の情報を取得します。ポーリング結果は、グラフ形式で表示されます。この結果を解析して、

インターネットワーキング関連の問題のトラブルシューティング、ネットワーク パフォーマンスの改善、デバイス設定の確認、トラフィック負荷のモニタなどを行うことができます。

次の図に示すように、SNMP エージェントは MIB からデータを収集します。エージェントは SNMP マネージャに対し、トラップ（特定イベントの通知）を送信でき、SNMP マネージャはトラップを受信して処理します。トラップは、ネットワーク上で発生した不正なユーザ認証、再起動、リンク ステータス（アップまたはダウン）、MAC アドレス トラッキングなどの状況を SNMP マネージャに通知します。SNMP エージェントはさらに、SNMP マネージャから *get-request*、*get-next-request*、および *set-request* 形式で送信される MIB 関連のクエリに応答します。

図 *2 : SNMP* ネットワーク



SNMP 通知

SNMP を使用すると、特定のイベントが発生した場合に、デバイスから SNMP マネージャに通知を送信できます。SNMP 通知は、トラップまたは情報要求として送信できます。コマンド構文では、トラップまたは情報を選択するオプションがコマンドにない限り、キーワード traps はトラップ、情報、またはその両方を表します。**snmp-server host** コマンドを使用して、トラップまたは情報として SNMP 通知を送信するかどうかを指定します。

（注） SNMPv1 は informs をサポートしていません。

トラップは信頼性に欠けます。受信側はトラップを受信しても確認応答を送信しないので、トラップが受信されたかどうかが送信側にわからないからです。情報要求の場合、受信した SNMP マネージャは SNMP 応答プロトコル データ ユニット（PDU）でメッセージを確認します。送信側が応答を受信しなかった場合は、再び情報要求を送信できます。再送信できるので、情報の方がトラップより意図した宛先に届く可能性が高くなります。

情報の方がトラップより信頼性が高いのは、デバイスおよびネットワークのリソースを多く消費するという特性にも理由があります。送信と同時に廃棄されるトラップと異なり、情報要求は応答を受信するまで、または要求がタイムアウトになるまで、メモリ内に保持されます。トラップの送信は 1 回限りですが、情報は数回にわたって再送信つまり再試行が可能です。再送信の回数が増えるとトラフィックが増加し、ネットワークのオーバーヘッドが高くなる原因にもなります。したがって、トラップにするか情報にするかは、信頼性を取るかリソースを取るかという選択になります。SNMP マネージャですべての通知を受信することが重要な場合は、情報要求を使用してください。ネットワークまたはデバイスのメモリ上のトラフィックが問題になる場合で、なおかつ通知が不要な場合は、トラップを使用してください。

SNMP ifIndex MIB オブジェクト値

NMS の IF-MIB は、物理インターフェイスまたは論理インターフェイスを識別する、ゼロより大きい一意の値である interface index（ifIndex）オブジェクト値の生成および割り当てを行います。デバイスの再起動またはデバイスソフトウェアのアップグレード時に、デバイスは、インターフェイスにこれと同じ値を使用します。たとえば、デバイスのポート 2 に 10003 という ifIndex 値が割り当てられていると、デバイスの再起動後も同じ値が使用されます。

デバイスは、次の表内のいずれかの値を使用して、インターフェイスに ifIndex 値を割り当てます。

表 *7 : ifIndex* 値

インターフェイスタイプ	ifIndex 範囲
SVI[4]	1 ～ 4999
EtherChannel	5001 ～ 5048
トンネル	5078 ～ 5142
タイプとポート番号に基づく物理（ギガビットイーサネットまたはSFP[5] モジュール インターフェイスなど）	10000 ～ 14500
ヌル	14501
ループバックおよびトンネル	24567+

[4] SVI = スイッチ仮想インターフェイス
[5] SFP = Small Form-Factor Pluggable

SNMP のデフォルト設定

機能	デフォルト設定
SNMP エージェント	ディセーブル[6]
SNMP トラップ レシーバ	未設定
SNMP トラップ	TCP接続のトラップ（tty）以外は、イネーブルではありません。
SNMP バージョン	バージョン キーワードがない場合、デフォルトはバージョン 1 になります。
SNMPv3 認証	キーワードを入力しなかった場合、セキュリティレベルはデフォルトで **noauth**（noAuthNoPriv）になります。
SNMP 通知タイプ	タイプが指定されていない場合、すべての通知が送信されます。

[6] これは、デバイスが起動し、スタートアップ コンフィギュレーションに **snmp-server** グローバル コンフィギュレーション コマンドが設定されていない場合のデフォルトです。

SNMP 設定時の注意事項

デバイスが起動し、デバイスのスタートアップ コンフィギュレーションに少なくとも 1 つの **snmp-server** グローバル コンフィギュレーション コマンドが設定されている場合、SNMP エージェントは有効になります。

SNMP グループは、SNMP ユーザを SNMP ビューに対応付けるテーブルです。SNMP ユーザは、SNMP グループのメンバです。SNMP ホストは、SNMP トラップ動作の受信側です。SNMP エンジン *ID* は、ローカルまたはリモート SNMP エンジンの名前です。

SNMP グループを設定するときには、次の注意事項に従ってください。

- SNMP グループを設定するときは、通知ビューを指定しません。**snmp-server host** グローバル コンフィギュレーション コマンドがユーザの通知ビューを自動生成し、そのユーザに関連付けられているグループに追加します。グループの通知ビューを変更すると、そのグループに対応付けられたすべてのユーザが影響を受けます。
- リモート ユーザを設定する場合は、ユーザが存在するデバイスのリモート SNMP エージェントに対応する IP アドレスまたはポート番号を指定します。
- 特定のエージェントのリモートユーザを設定する前に、**snmp-server engineID** グローバル コンフィギュレーション コマンドを **remote** オプションとともに使用して、SNMP エンジン ID を設定してください。リモート エージェントの SNMP エンジン ID およびユーザ パスワードを使用して認証およびプライバシーダイジェストが算出されます。先にリモート エンジン ID を設定しておかないと、コンフィギュレーション コマンドがエラーになります。
- SNMP 情報を設定するときには、プロキシ要求または情報の送信先となるリモート エージェントの SNMP エンジン ID を SNMP データベースに設定しておく必要があります。
- ローカルユーザがリモートホストと関連付けられていない場合、デバイスは **auth**（authNoPriv）および **priv**（authPriv）の認証レベルの情報を送信しません。
- SNMP エンジン ID の値を変更すると、重大な影響が生じます。（コマンド ラインで入力された）ユーザのパスワードは、パスワードおよびローカル エンジン ID に基づいて、MD5 または SHA セキュリティ ダイジェストに変換されます。コマンドラインのパスワードは、RFC 2274 の規定に従って廃棄されます。このようにパスワードが廃棄されるため、エンジン ID 値を変更した場合は、SNMPv3 ユーザのセキュリティダイジェストが無効となり、**snmp-server user** *username* グローバル コンフィギュレーション コマンドを使用して、SNMP ユーザを再設定する必要があります。エンジン ID を変更した場合は、同様の制限によってコミュニティ ストリングも再設定する必要があります。

SNMP の設定方法

ここでは、SNMP の設定について説明します。

SNMP エージェントのディセーブル化

no snmp-server グローバル コンフィギュレーション コマンドは、デバイス上で実行している SNMP エージェントのすべてのバージョン（バージョン 1、バージョン 2C、バージョン 3）をディセーブルにします。入力した最初の **snmp-server** グローバル コンフィギュレーション コマンドによって、SNMP エージェントのすべてのバージョンを再度イネーブルにします。特に SNMP をイネーブルにするために指定された Cisco IOS コマンドはありません。

SNMP エージェントをディセーブルにするには、次の手順を実行します。

始める前に

SNMP エージェントをディセーブルにする前にイネーブルにする必要があります。デバイス上で入力した最初の **snmp-server** グローバル コンフィギュレーション コマンドによって SNMP エージェントがイネーブルになります。

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。 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	**no snmp-server** 例： `Device(config)# no snmp-server`	SNMP エージェント動作をディセーブルにします。
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： `Device# show running-config`	入力を確認します。

	コマンドまたはアクション	目的
ステップ 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。

コミュニティ ストリングの設定

SNMPマネージャとエージェントの関係を定義するには、SNMPコミュニティストリングを使用します。コミュニティストリングは、デバイス上のエージェントへのアクセスを許可する、パスワードと同様の役割を果たします。ストリングに対応する次の特性を1つまたは複数指定することもできます。

- コミュニティ ストリングを使用してエージェントにアクセスできる SNMP マネージャの IP アドレスのアクセス リスト
- 指定のコミュニティにアクセスできるすべての MIB オブジェクトのサブセットを定義する MIB ビュー
- コミュニティにアクセスできる MIB オブジェクトの読み書き権限または読み取り専用権限

デバイス上でコミュニティストリングを設定するには、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	**snmp-server community** *string*［**view** *view-name*］［**ro** | **rw**］［*access-list-number*］ 例： `Device(config)# snmp-server community comaccess ro 4`	コミュニティ ストリングを設定します。 （注） コンテキスト情報を区切るには @ 記号を使用します。このコマンドの設定時に SNMP コミュニティ ストリングの一部として @ 記号を使用しないでください。 • *string* には、パスワードと同様に機能し、SNMP プロトコルへのアクセスを許可するストリングを指定します。任意の長さのコミュニティ ストリングを 1 つまたは複数設定できます。 •（任意）**view** には、コミュニティがアクセスできるビューレコードを指定します。 •（任意）許可された管理ステーションで MIB オブジェクトを取得する場合は読み取り専用（**ro**）、許可された管理ステーションで MIB オブジェクトを取得および変更する場合は読み書き（**rw**）を指定します。デフォルトでは、コミュニティ ストリングはすべてのオブジェクトに対する読み取り専用アクセスを許可します。 •（任意）*access-list-number* には、1 ～ 99 および 1300 ～ 1999 の標準 IP アクセスリスト番号を入力します。
ステップ 4	**access-list** *access-list-number* {**deny** | **permit**} *source*［*source-wildcard*］ 例： `Device(config)# access-list 4 deny any`	（任意）ステップ 3 で標準 IP アクセスリスト番号を指定してリストを作成した場合は、必要に応じてコマンドを繰り返します。 • *access-list-number* には、ステップ 3 で指定したアクセス リスト番号を入力します。 • **deny** キーワードは、条件が一致した場合にアクセスを拒否します。

	コマンドまたはアクション	目的
		permit キーワードは、条件が一致した場合にアクセスを許可します。 • *source* には、コミュニティ ストリングを使用してエージェントにアクセスできる SNMP マネージャの IP アドレスを入力します。 • （任意）*source-wildcard* には、source に適用されるワイルドカードビットをドット付き 10 進表記で入力します。無視するビット位置には 1 を設定します。 アクセス リストの末尾には、すべてに対する暗黙の拒否ステートメントが常に存在することに注意してください。
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。

次のタスク

SNMP コミュニティのアクセスをディセーブルにするには、そのコミュニティのコミュニティストリングをヌル ストリングに設定します（コミュニティ ストリングに値を入力しないでください）。

特定のコミュニティストリングを削除するには、**no snmp-server** 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。

デバイスのローカルまたはリモート SNMP サーバエンジンを表す識別名（エンジン ID）を指定できます。SNMP ユーザを SNMP ビューにマッピングする、SNMP サーバ グループを設定し、新規ユーザを SNMP グループに追加できます。

SNMP グループおよびユーザの設定

デバイスのローカルまたはリモート SNMP サーバエンジンを表す識別名（エンジン ID）を指定できます。SNMP ユーザを SNMP ビューにマッピングする、SNMP サーバ グループを設定し、新規ユーザを SNMP グループに追加できます。

デバイス上の SNMP グループとユーザを設定するには、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	**snmp-server engineID** {**local** *engineid-string* | **remote** *ip-address* [**udp-port** *port-number*] *engineid-string*} 例： `Device(config)# snmp-server engineID local 1234`	SNMP のローカル コピーまたはリモート コピーに名前を設定します。 • *engineid-string* は、SNMP のコピー名を指定する 24 文字の ID ストリングです。後続ゼロが含まれる場合は、24 文字のエンジン ID すべてを指定する必要はありません。指定するのは、エンジン ID のうちゼロのみが続く箇所を除いた部分だけです。手順例では、123400000000000000000000 のエンジン ID を設定します。 • **remote** を指定した場合、SNMP のリモートコピーが置かれているデバイスの *ip-address* を指定し、任意でリモートデバイスのユーザ データグラム プロトコル（UDP）ポート

	コマンドまたはアクション	目的
		を指定します。デフォルトは 162 です。
ステップ 4	**snmp-server group** *group-name* {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}} [**read** *readview*] [**write** *writeview*] [**notify** *notifyview*] [**access** *access-list*] 例： `Device(config)# snmp-server group public v2c access lmnop`	リモート デバイス上で新しい SNMP グループを設定します。 *group-name* には、グループの名前を指定します。 次のいずれかのセキュリティ モデルを指定します。 • **v1** は、最も安全性の低いセキュリティ モデルです。 • **v2c** は、2 番目に安全性の低いセキュリティ モデルです。標準の 2 倍の幅で情報および整数を伝送できます。 • **v3**最も安全な場合には、次の認証レベルの 1 つを選択する必要があります。 **auth**：Message Digest 5（MD5）およびセキュア ハッシュ アルゴリズム（SHA）によるパケット認証を可能にします。 **noauth**：noAuthNoPriv セキュリティ レベルを可能にします。キーワードを指定しなかった場合、これがデフォルトです。 **priv**：データ暗号規格（DES）によるパケット暗号化（プライバシーともいう）を可能にします。 （任意）**read** *readview* とともに、エージェントの内容を表示できるビュー名を表す文字列（64 文字以内）を入力します。 （任意）**write** *writeview* とともに、データを入力し、エージェントの内容を表示できるビュー名を表す文字列（64 文字以内）を入力します。

	コマンドまたはアクション	目的
		（任意）**notify** *notifyview* とともに、通知、情報、またはトラップを指定するビュー名を表す文字列（64 文字以内）を入力します。 （任意）**access** *access-list* とともに、アクセスリスト名の文字列（64 文字以内）を入力します。
ステップ 5	**snmp-server user** *username group-name* {**remote** *host* [**udp-port** *port*] } {**v1** [**access** *access-list*] | **v2c** [**access** *access-list*] | **v3** [**encrypted**] [**access** *access-list*] [**auth** {**md5** | **sha**} *auth-password*] } [*priv* {**des** | **3des** | **aes** {**128** | **192** | **256**} } *priv-password*] 例： `Device(config)# snmp-server user Pat public v2c`	SNMP グループに対して新規ユーザを追加します。 *username* は、エージェントに接続するホスト上のユーザ名です。 *group-name* は、ユーザが関連付けられているグループの名前です。 **remote** を入力して、ユーザが所属するリモート SNMP エンティティおよびそのエンティティのホスト名または IP アドレスとともに、任意で UDP ポート番号を指定します。デフォルトは 162 です。 SNMP バージョン番号（**v1**、**v2c**、または **v3**）を入力します。**v3** を入力する場合は、次のオプションを追加します。 • **encrypted** は、パスワードを暗号化形式で表示するように指定します。このキーワードは、**v3** キーワードが指定されている場合にのみ使用できます。 • **auth** では、認証レベルを設定します。 HMAC-MD5-96（**md5**）または HMAC-SHA-96（**sha**）認証レベルを指定できます。また、*auth-password* でパスワードの文字列を指定する必要があります（最大 64 文字）。 **v3** を入力すると、次のキーワードを使用して（64 文字以内）、プライベート（**priv**）暗号化アルゴリズムおよびパスワード文字列 *priv-password* を設定することもできます。

	コマンドまたはアクション	目的
		• **priv** は、ユーザベース セキュリティ モデル（USM）を指定します。 • **des** 56 ビット DES アルゴリズムを使用する場合に指定します。 • **3des** 168 ビット DES アルゴリズムを使用する場合に指定します。 • **aes** DES アルゴリズムを使用する場合に指定します。128 ビット暗号化、192 ビット暗号化、または 256 ビット暗号化のいずれかを選択する必要があります。 （任意）**access** *access-list* とともに、アクセスリスト名の文字列（64 文字以内）を入力します。
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 8	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

SNMP 通知の設定

トラップマネージャは、トラップを受信して処理する管理ステーションです。トラップは、特定のイベントが発生したときに装置が生成するシステム アラートです。デフォルトでは、トラップ マネージャは定義されず、トラップは送信されません。この Cisco IOS リリースが稼働しているデバイスでは、トラップマネージャを無制限に設定できます。

（注） コマンド構文で **traps** というワードを使用するコマンドは多数あります。トラップまたは情報を選択するオプションがコマンドにない限り、キーワード**traps**はトラップ、情報のいずれか、またはその両方を表します。**snmp-server host** グローバル コンフィギュレーション コマンドを使用して、トラップまたは情報として SNMP 通知を送信するかどうかを指定します。

次の表に示す通知タイプを受信するには、特定のホストに対して **snmp-server host** グローバル コンフィギュレーション コマンドを使用します。これらのトラップの一部または全部をイネーブルにして、これを受信するようにトラップ マネージャを設定できます。

表 *8 :* デバイスの通知タイプ

通知タイプのキーワード	説明
bridge	STP ブリッジ MIB トラップを生成します。
cluster	クラスタ設定が変更された場合に、トラップを生成します。
config	SNMP 設定が変更された場合に、トラップを生成します。
copy-config	SNMP コピー設定が変更された場合に、トラップを生成します。
cpu threshold	CPU に関連したトラップをイネーブルにします。
entity	SNMP エンティティが変更された場合に、トラップを生成します。
envmon	環境モニタ トラップを生成します。ファン（fan）、シャットダウン（shutdown）、ステータス（status）、電源（supply）、温度（temperature）の環境トラップのいずれかまたはすべてをイネーブルにできます。
errdisable	ポート VLAN が errdisable ステートになった場合に、トラップを生成します。1 分あたりの最大トラップ速度も設定できます。指定できる範囲は 0 ～ 10000 です。デフォルトは 0 で、レート制限がないという意味です。
flash	SNMP FLASH 通知を生成します。スイッチ スタックでは、オプションとして、フラッシュの追加または削除に関する通知をイネーブルにできます。このようにすると、スタックからスイッチを削除するか、またはスタックにスイッチを追加した場合に（物理的な取り外し、電源のオフ/オン、またはリロードの場合に）、トラップが発行されます。
fru-ctrl	エンティティ現場交換可能ユニット（FRU）制御トラップを生成します。スイッチ スタックでは、このトラップはスタックにおけるスイッチの挿入/取り外しを意味します。

通知タイプのキーワード	説明
hsrp	ホットスタンバイ ルータ プロトコル（HSRP）が変更された場合に、トラップを生成します。
ipmulticast	IP マルチキャスト ルーティングが変更された場合に、トラップを生成します。
ipsla	SNMP IP サービスレベル契約（SLA）のトラップを生成します。
mac-notification	MAC アドレス通知のトラップを生成します。
msdp	Multicast Source Discovery Protocol（MSDP）が変更された場合に、トラップを生成します。
ospf	Open Shortest Path First（OSPF）が変更された場合に、トラップを生成します。シスコ固有、エラー、リンクステートアドバタイズ、レート制限、再送信、ステート変更に関するトラップを任意にイネーブルにできます。
pim	Protocol-Independent Multicast（PIM）が変更された場合に、トラップを生成します。無効な PIM メッセージ、ネイバー変更、およびランデブー ポイント（RP）マッピングの変更に関するトラップを任意にイネーブルにできます。
port-security	SNMP ポート セキュリティ トラップを生成します。1 秒あたりの最大トラップ速度も設定できます。指定できる範囲は 0 ～ 1000 秒です。デフォルトは 0 秒で、レート制限がないという意味です。 （注） 通知タイプ **port-security** を使用してトラップを設定する際に、まずポートセキュリティトラップを設定して、次に以下のポートセキュリティ トラップ レートを設定します。 **1. snmp-server enable traps port-security** **2. snmp-server enable traps port-security trap-rate** *rate*
snmp	認証、コールドスタート、ウォームスタート、リンクアップ、またはリンク ダウンについて、SNMP タイプ通知のトラップを生成します。
storm-control	SNMP ストーム制御のトラップを生成します。1 分あたりの最大トラップ速度も設定できます。指定できる範囲は 0 ～ 1000 です。デフォルトは 0 に設定されています（制限なしの状態では、発生ごとにトラップが送信されます）。
stpx	SNMP STP 拡張 MIB トラップを生成します。
syslog	SNMP の Syslog トラップを生成します。

通知タイプのキーワード	説明
tty	TCP 接続のトラップを生成します。このトラップは、デフォルトでイネーブルに設定されています。
vlan-membership	SNMP VLAN メンバーシップが変更された場合に、トラップを生成します。
vlancreate	SNMP VLAN 作成トラップを生成します。
vlandelete	SNMP VLAN 削除トラップを生成します。
vtp	VLAN トランキング プロトコル（VTP）が変更された場合に、トラップを生成します。

ホストにトラップまたは情報を送信するようにデバイスを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • 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	**snmp-server engineID remote** *ip-address engineid-string* 例： `Device(config)# snmp-server engineID remote 192.180.1.27 00000063000100a1c0b4011b`	リモート ホストのエンジン ID を指定します。
ステップ 4	**snmp-server user** *username group-name* {**remote** *host* [**udp-port** *port*] } {**v1** [**access** *access-list*] | **v2c** [**access** *access-list*] | **v3** [**encrypted**] [**access**	SNMP ユーザを設定し、ステップ 3 で作成したリモート ホストに関連付けます。

	コマンドまたはアクション	目的
	access-list］［**auth**｛**md5**	**sha**｝*auth-password*］｝ 例： `Device(config)# snmp-server user Pat public v2c`	（注） アドレスに対応するリモートユーザを設定するには、先にリモート ホストのエンジン ID を設定しておく必要があります。このようにしないと、エラーメッセージが表示され、コマンドが実行されません。
ステップ 5	**snmp-server group** *group-name*｛**v1**	**v2c**	**v3**｛**auth**	**noauth**	**priv**｝｝［**read** *readview*］［**write** *writeview*］［**notify** *notifyview*］［**access** *access-list*］ 例： `Device(config)# snmp-server group public v2c access lmnop`	SNMP グループを設定します。
ステップ 6	**snmp-server host** *host-addr*［**informs**	**traps**］［**version**｛**1**	**2c**	**3**｛**auth**	**noauth**	**priv**｝｝］*community-string*［*notification-type*］ 例： `Device(config)# snmp-server host 203.0.113.1 comaccess snmp`	SNMP トラップ動作の受信先を指定します。 *host-addr* には、ホスト（対象となる受信側）の名前またはインターネットアドレスを指定します。 （任意）SNMP トラップをホストに送信するには、**traps**（デフォルト）を指定します。 （任意）SNMP 情報をホストに送信するには、**informs** を指定します。 （任意）SNMP **version**（**1** 、**2c** 、または **3**）を指定します。SNMPv1 は informs をサポートしていません。 （任意）バージョン 3 の場合、認証レベルとして **auth**、**noauth**、または **priv** を選択します。 （注） **priv** キーワードは、暗号化ソフトウェアイメージがインストールされている場合のみ指定できます。 *community-string* には、**version 1** または **version 2c** が指定されている場合、通知動作で送信される、パスワードに類似したコミュニティストリングを入力し

	コマンドまたはアクション	目的
		ます。**version 3** が指定されている場合は、SNMPv3 のユーザ名を入力します。 コンテキスト情報を区切るには@記号を使用します。このコマンドの設定時にSNMPコミュニティストリングの一部として@記号を使用しないでください。 （任意）*notification-type* には、上の表に記載されているキーワードを使用します。タイプが指定されていない場合、すべての通知が送信されます。
ステップ 7	**snmp-server enable traps** *notification-types* 例： `Device(config)# snmp-server enable traps snmp`	デバイスでのトラップまたはインフォームの送信をイネーブルにし、送信する通知の種類を指定します。通知タイプの一覧については、上の表を参照するか、と入力してください。**snmp-server enable traps ?** 複数のトラップタイプをイネーブルにするには、トラップタイプごとに **snmp-server enable traps** コマンドを個別に入力する必要があります。 （注） 通知タイプ **port-security** を使用してトラップを設定する際に、まずポートセキュリティトラップを設定して、次に以下のポート セキュリティ トラップ レートを設定します。 1. **snmp-server enable traps port-security** 2. **snmp-server enable traps port-security trap-rate** *rate*
ステップ 8	**snmp-server trap-source** *interface-id* 例： `Device(config)# snmp-server trap-source gigabitethernet 1/0/1`	（任意）送信元インターフェイスを指定します。このインターフェイスによってトラップ メッセージの IP アドレスが提供されます。情報の送信元 IP アドレスも、このコマンドで設定します。

	コマンドまたはアクション	目的
ステップ 9	**snmp-server queue-length** *length* 例： `Device(config)# snmp-server queue-length 20`	（任意）各トラップホストのメッセージキューの長さを指定します。指定できる値の範囲は 1 ～ 5000 です。デフォルトは 10 です。
ステップ 10	**snmp-server trap-timeout** *seconds* 例： `Device(config)# snmp-server trap-timeout 60`	（任意）トラップメッセージを再送信する頻度を指定します。指定できる範囲は 1 ～ 1000 です。デフォルトは 30 秒です。
ステップ 11	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 12	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 13	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 14	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーションファイルに設定を保存します。

次のタスク

snmp-server host コマンドでは、通知を受信するホストを指定します。**snmp-server enable traps** コマンドによって、指定された通知方式（トラップおよび情報）がグローバルにイネーブルになります。ホストが情報を受信できるようにするには、そのホストに対応する **snmp-server host informs** コマンドを設定し、**snmp-server enable traps** コマンドを使用して情報をグローバルにイネーブルにする必要があります。

指定したホストがトラップを受信しないようにするには、 **no snmp-server host** *host* 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。キーワードを指定しないで **no snmp-server host** コマンドを使用すると、ホストへのトラップはディセーブルになりますが、情報はディセーブルになりません。情報をディセーブルにするには、**no snmp-server host informs** グロー

バル コンフィギュレーション コマンドを使用します。特定のトラップタイプをディセーブルにするには、**no snmp-server enable traps** *notification-types* 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。

エージェント コンタクトおよびロケーションの設定

SNMPエージェントのシステム接点およびロケーションを設定して、コンフィギュレーション ファイルからこれらの記述にアクセスできるようにするには、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	**snmp-server contact** *text* 例： `Device(config)# snmp-server contact Dial System Operator at beeper 21555`	システムの連絡先文字列を設定します。
ステップ 4	**snmp-server location** *text* 例： `Device(config)# snmp-server location Building 3/Room 222`	システムの場所を表す文字列を設定します。
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。

	コマンドまたはアクション	目的
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

SNMP を通して使用する TFTP サーバの制限

SNMP を介したコンフィギュレーション ファイルの保存とロードに使用する TFTP サーバを、アクセス リストで指定されたサーバに限定するには、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	**snmp-server tftp-server-list** *access-list-number* 例： `Device(config)# snmp-server tftp-server-list 44`	SNMP を介したコンフィギュレーション ファイルのコピーに使用する TFTP サーバを、アクセス リストのサーバに限定します。 *access-list-number* には、1 ～ 99 および 1300 ～ 1999 の標準 IP アクセス リスト番号を入力します。
ステップ 4	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] 例： `Device(config)# access-list 44 permit 10.1.1.2`	標準アクセス リストを作成し、コマンドを必要な回数だけ実行します。 *access-list-number* には、ステップ 3 で指定したアクセス リスト番号を入力します。 **deny** キーワードは、条件が一致した場合にアクセスを拒否します。**permit** キー

	コマンドまたはアクション	目的
		ワードは、条件が一致した場合にアクセスを許可します。 *source* には、デバイスにアクセスできる TFTP サーバの IP アドレスを入力します。 （任意）*source-wildcard* には、source に適用されるワイルドカードビットをドット付き 10 進表記で入力します。無視するビット位置には 1 を設定します。 アクセス リストの末尾には、すべてに対する暗黙の拒否ステートメントが常に存在します。
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。

SNMP ステータスのモニタリング

不正なコミュニティ ストリング エントリ、エラー、要求変数の数など、SNMP の入出力統計情報を表示するには、**show snmp** 特権 EXEC コマンドを使用します。また、次の表にリストされたその他の特権 EXEC コマンドを使用して、SNMP 情報を表示することもできます。

表 *9 : SNMP* 情報を表示するためのコマンド

コマンド	目的
show snmp	SNMP 統計情報を表示します。

コマンド	目的
show snmp group	ネットワーク上の各 SNMP グループに関する情報を表示します。
show snmp pending	保留中の SNMP 要求の情報を表示します。
show snmp sessions	現在の SNMP セッションの情報を表示します。
show snmp user	SNMP ユーザ テーブルの各 SNMP ユーザ名に関する情報を表示します。 （注） このコマンドは、**auth** | **noauth** | **priv** モードの SNMPv3 設定情報を表示する際に使用する必要があります。この情報は、**show running-config** の出力には表示されません。

SNMP の例

次に、SNMP の全バージョンをイネーブルにする例を示します。この設定では、任意の SNMP マネージャがコミュニティ ストリング *public* を使用して、読み取り専用権限ですべてのオブジェクトにアクセスできます。この設定では、デバイスはトラップを送信しません。

```
Device(config)# snmp-server community public
```

次に、任意の SNMP マネージャがコミュニティ ストリング *public* を使用して、読み取り専用権限ですべてのオブジェクトにアクセスする例を示します。デバイスはさらに、SNMPv1 を使用してホスト 192.180.1.111 および 192.180.1.33 に、SNMPv2C を使用してホスト 192.180.1.27 に VTP トラップを送信します。コミュニティ ストリング *public* は、トラップとともに送信されます。

```
Device(config)# snmp-server community public
Device(config)# snmp-server enable traps vtp
Device(config)# snmp-server host 192.180.1.27 version 2c public
Device(config)# snmp-server host 192.180.1.111 version 1 public
Device(config)# snmp-server host 192.180.1.33 public
```

次に、*comaccess* コミュニティ ストリングを使用するアクセス リスト 4 のメンバに、すべてのオブジェクトへの読み取り専用アクセスを許可する例を示します。その他の SNMP マネージャは、どのオブジェクトにもアクセスできません。SNMP 認証障害トラップは、SNMPv2C がコミュニティ ストリング *public* を使用してホスト *cisco.com* に送信します。

```
Device(config)# snmp-server community comaccess ro 4
Device(config)# snmp-server enable traps snmp authentication
Device(config)# snmp-server host cisco.com version 2c public
```

次に、エンティティ MIB トラップをホスト *cisco.com* に送信する例を示します。コミュニティ ストリングは制限されます。1 行目で、デバイスはすでにイネーブルになっているトラップ以外に、エンティティ MIB トラップを送信できるようになります。2 行目はこれらのトラップの宛先を指定し、ホスト *cisco.com* に対する以前の **snmp-server** ホストコマンドを無効にします。

```
Device(config)# snmp-server enable traps entity
Device(config)# snmp-server host cisco.com restricted entity
```

次に、コミュニティストリング *public* を使用して、すべてのトラップをホスト *myhost.cisco.com* に送信するようにデバイスをイネーブルにする例を示します。

```
Device(config)# snmp-server enable traps
Device(config)# snmp-server host myhost.cisco.com public
```

次に、ユーザとリモートホストを関連付けて、ユーザがグローバル コンフィギュレーション モードの際に **auth**（authNoPriv）認証レベルで情報を送信する例を示します。

```
Device(config)# snmp-server engineID remote 192.180.1.27 00000063000100a1c0b4011b
Device(config)# snmp-server group authgroup v3 auth
Device(config)# snmp-server user authuser authgroup remote 192.180.1.27 v3 auth md5
mypassword
Device(config)# snmp-server user authuser authgroup v3 auth md5 mypassword
Device(config)# snmp-server host 192.180.1.27 informs version 3 auth authuser config
Device(config)# snmp-server enable traps
Device(config)# snmp-server inform retries 0
```

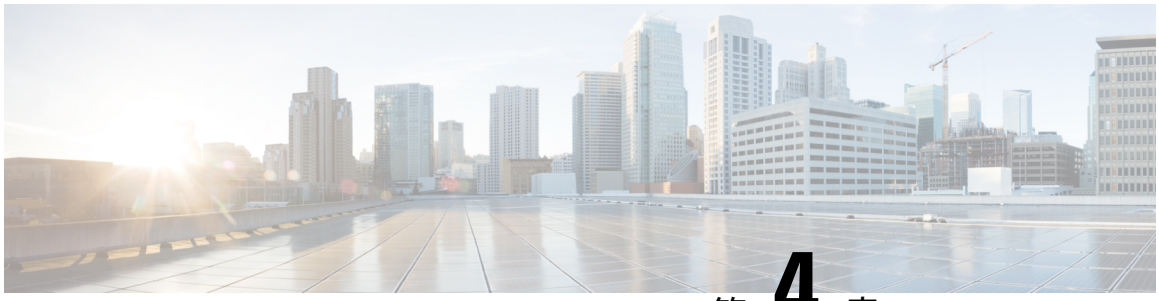
SNMP の設定に関する機能情報

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 *10 : SNMP* の設定に関する機能情報

機能名	リリース	機能情報
SNMP の設定	Cisco IOS Release 15.2(7)E1	この機能が導入されました。



第 4 章

SPAN の設定

SPAN の制約事項

SPAN

SPAN の制約事項は次のとおりです。

- SPAN 送信元の場合は、セッションごとに、単一のポート、一連のポート、一定範囲のポートのトラフィックをモニタできます。
- 宛先ポートを送信元ポートにすることはできません。同様に、送信元ポートを宛先ポートにすることもできません。
- 同じ宛先ポートで 2 つの SPAN セッションを設定することはできません。
- 同じ送信元ポートで 2 つの SPAN セッションを設定することはできません。
- スイッチ ポートを SPAN 宛先ポートとして設定すると、通常のスイッチ ポートではなくなります。SPAN 宛先ポートを通過するトラフィックがモニタされるだけです。
- SPAN コンフィギュレーション コマンドを入力しても、前に設定した SPAN パラメータは削除されません。設定されている SPAN パラメータを削除するには、**no monitor session** *session_number* 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。

- 無効のポートを送信元ポートまたは宛先ポートとして設定することはできますが、SPAN 機能が開始されるのは、宛先ポートと少なくとも 1 つの送信元ポートが有効になってからです。

SPAN セッションのトラフィック監視には次の制約事項があります。

- デバイスは、最大 4 つのローカル SPAN セッションをサポートします。
- SPAN セッションがデバイスの通常の動作を妨げることはありません。ただし、10 Mbps のポートで 100 Mbps のポートをモニタするなど、オーバーサブスクライブの SPAN 宛先は、パケットのドロップまたは消失を招くことがあります。
- SPAN が有効な場合、監視中の各パケットは 2 回送信されます（1 回は標準トラフィックとして、もう 1 回は監視されたパケットとして）。多数のポートをモニタリングすると、大量のネットワーク トラフィックが生成されることがあります。
- 無効のポート上に SPAN セッションを設定することはできますが、そのセッション用に宛先ポートと少なくとも 1 つの送信元ポートを有効にしない限り、SPAN セッションはアクティブになりません。

SPAN について

ここでは、SPAN について説明します。

SPAN

ポートを通過するネットワーク トラフィックを解析するには、SPAN を使用して、そのスイッチ上の別のポート、またはネットワーク アナライザやその他のモニタ デバイスもしくはセキュリティ デバイスに接続されている別のスイッチ上のポートに、トラフィックのコピーを送信します。SPAN は送信元ポート上で受信、送信、または送受信されたトラフィックを宛先ポートにコピー（ミラーリング）して、解析します。SPAN は発信元ポート上のネットワーク トラフィックのスイッチングには影響しません。宛先ポートは SPAN 専用にする必要があります。SPAN セッションに必要なトラフィック以外、宛先ポートがトラフィックを受信したり転送したりすることはありません。

SPAN を使用してモニタできるのは、送信元ポートを出入りするトラフィックだけです。

ネットワーク セキュリティ デバイスからトラフィックを注入する場合、SPAN 宛先ポートを使用できます。たとえば、Cisco 侵入検知システム（IDS）センサー装置を宛先ポートに接続した場合、IDS デバイスは TCP リセット パケットを送信して、疑わしい攻撃者の TCP セッションを停止させることができます。

SPAN のデフォルト設定

表 *11 : SPAN* のデフォルト設定

機能	デフォルト設定
SPAN ステート	ディセーブル
モニタする送信元ポート トラフィック	受信トラフィックと送信トラフィックの両方（**both**）
カプセル化タイプ（宛先ポート）	ネイティブ形式（タグなしパケット）
入力転送（宛先ポート）	ディセーブル

SPAN 設定時の注意事項

- SPANセッションから送信元ポートまたは宛先ポートを削除する場合は、**no monitor session** *session_number* **source interface** *interface-id* グローバル コンフィギュレーション コマンドまたは **no monitor session** *session_number* **destination interface** *interface-id* グローバル コンフィギュレーション コマンドを使用します。宛先インターフェイスの場合、このコマンドの **no** 形式を使用すると、**encapsulation** オプションは無視されます。

SPAN の設定方法

ここでは、SPAN の設定方法について説明します。

ローカル SPAN セッションの作成

SPAN セッションを作成し、送信元（監視対象）ポートまたは VLAN、および宛先（監視側）ポートを指定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例：	グローバル 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	**no monitor session** *session_number* 例： `Device(config)# no monitor session 1`	特定のセッションに対する既存の SPAN 設定を削除します。指定できる範囲は 1 ～ 4 です。

ステップ 4	**monitor session** *session_number* **source** { **interface** *interface-id*} [**,**	**-**] [**both**	**rx**	**tx**]
				例：
				`Device(config)# monitor session 1 source interface gigabitethernet 1/0/1`
					SPAN セッションおよび送信元ポート（監視対象ポート）を指定します。
					• *session_number* の範囲は、1 ～ 4 です。
					• *interface-id* には、監視する送信元ポートを指定します。有効なインターフェイスには、物理インターフェイスおよびポートチャネル論理インターフェイス（**port-channel** *port-channel-number*）があります。有効なポートチャネル番号は 1 ～ 6 です。
					•（任意）**[,|-]** には、一連のインターフェイスまたはインターフェイスの範囲を指定します。カンマの前後およびハイフンの前後にスペースを 1 つずつ入力します。
					•（任意）**both**
						rx	**tx**：監視するトラフィックの方向を指定します。トラフィックの方向を指定しなかった場合、送信元インターフェイスは送信トラフィックと受信トラフィックの両方を送信します。
							  • **both**：受信トラフィックと送信トラフィックの両方を監視します。
							  • **rx**：受信トラフィックをモニタします。
							  • **tx**：送信トラフィックをモニタします。

	コマンドまたはアクション	目的
		（注） **monitor session** *session_number***source** コマンドを複数回使用すると、複数の送信元ポートを設定できます。
ステップ 5	**monitor session** *session_number* **destination** { **interface** *interface-id* } 例： `Device(config)# monitor session 1 destination interface gigabitethernet 1/0/2`	SPAN セッションおよび宛先ポート（監視側ポート）を指定します。設定変更が有効になると、ポートの LED がオレンジ色に変わります。LED は SPAN 宛先の設定を削除した後にのみ、元の状態（緑色）に戻ります。 （注） ローカル SPAN の場合は、送信元および宛先インターフェイスに同じセッション番号を使用する必要があります。 • *session_number* には、ステップ 4 で入力したセッション番号を指定します。 • *interface-id* には、宛先ポートを指定します。宛先インターフェイスには物理ポートを指定する必要があります。EtherChannel や VLAN は指定できません。
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

ローカル SPAN セッションの作成および着信トラフィックの設定

SPAN セッションを作成し、さらに送信元ポートまたは VLAN および宛先ポートを指定した後、宛先ポートでネットワーク セキュリティ デバイス（Cisco IDS センサー装置等）用に着信トラフィックをイネーブルにするには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**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	**no monitor session** *session_number* 例： `Device(config)# no monitor session 1`	特定のセッションに対する既存の SPAN 設定を削除します。指定できる範囲は 1 ～ 4 です。
ステップ 4	**monitor session** *session_number* **source** { **interface** *interface-id*} [**,** | **-**] [**both** | **rx** | **tx**] 例： `Device(config)# monitor session 2 source gigabitethernet 1/0/1 rx`	SPAN セッションおよび送信元ポート（監視対象ポート）を指定します。
ステップ 5	**monitor session** *session_number* **destination** { **interface** *interface-id* [**encapsulation replicate ingress** { **vlan** *vlan-id*} | **ingress** { **vlan** *vlan-id*}]} 例： `Device(config)# monitor session 2 destination interface gigabitethernet 1/0/2 ingress vlan 6`	SPAN セッション、宛先ポート、パケットカプセル化、および入力 VLAN とカプセル化を指定します。 • *session_number* には、ステップ 4 で入力したセッション番号を指定します。 • *interface-id* には、宛先ポートを指定します。宛先インターフェイスには物理ポートを指定する必要があります。EtherChannel や VLAN は指定できません。

	コマンドまたはアクション	目的
		• （任意）**encapsulation replicate**：宛先インターフェイスが送信元インターフェイスのカプセル化方式を複製することを指定します。選択しない場合のデフォルトは、ネイティブ形式（タグなし）でのパケットの送信です。 • **ingress**：宛先ポートでの着信トラフィックの転送をイネーブルにして、カプセル化タイプを指定します。
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

SPAN 動作のモニタリング

次の表で、SPAN 動作の設定と結果を表示して動作をモニタするために使用するコマンドについて説明します。

表 *12 : SPAN* 動作のモニタリング

コマンド	目的
show monitor session	現在の SPAN 設定を表示します。 すべての SPAN セッションの設定を表示するにはキーワード **all**、ローカルセッションのみの設定を表示するにはキーワード **local**、ある範囲の SPAN セッションの設定を表示するにはキーワード **range** を、それぞれ入力します。

SPAN の設定例

例：ローカル SPAN の設定

次に、SPAN セッション 1 を設定し、宛先ポートへ向けた送信元ポートのトラフィックをモニタする例を示します。最初に、セッション 1 の既存の SPAN 設定を削除し、カプセル化方式を

維持しながら、双方向トラフィックを送信元ポート GigabitEthernet 1 から宛先ポート GigabitEthernet 2 にミラーリングします。

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 1
Device(config)# monitor session 1 source interface gigabitethernet 1/0/1
Device(config)# monitor session 1 destination interface gigabitethernet 1/0/2
encapsulation replicate
Device(config)# end
```

次に、SPAN セッション 1 の SPAN 送信元としてのポート 1 を削除する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 1 source interface gigabitethernet 1/0/1
Device(config)# end
```

次に、双方向モニタが設定されていたポート 1 で、受信トラフィックのモニタをディセーブルにする例を示します。

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 1 source interface gigabitethernet 1/0/1 rx
```

ポート 1 で受信するトラフィックのモニタはディセーブルになりますが、このポートから送信されるトラフィックは引き続きモニタされます。

次に、SPAN セッション 2 内の既存の設定を削除し、VLAN 1 ～ 3 に属するすべてのポートで受信トラフィックをモニタするように SPAN セッション 2 を設定し、モニタされたトラフィックを宛先ポート GigabitEthernet 2 に送信する例を示します。さらに、この設定は VLAN 10 に属するすべてのポートですべてのトラフィックをモニタするよう変更されます。

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 2
Device(config)# monitor session 2 destination interface gigabitethernet 1/0/2
Device(config)# end
```

次に、SPAN セッション 2 の既存の設定を削除し、ギガビット イーサネット ソース送信元ポート 1 上で受信されるトラフィックをモニタするように SPAN セッション 2 を設定し、そのトラフィックを送信元ポートと同じ出力カプセル化方式の宛先ギガビット イーサネット ポート 2 に送信し、デフォルト入力 VLAN として VLAN 6 を使用した入力転送をイネーブルにする例を示します。

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 2
Device(config)# monitor session 2 source gigabitethernet 1/0/1 rx
Device(config)# monitor session 2 destination interface gigabitethernet 1/0/2 encapsulation
 replicate ingress vlan 6
Device(config)# end
```

SPAN の機能履歴と情報

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

機能名	リリース	機能情報
スイッチ ポート アナライザ（SPAN）	Cisco IOS Release 15.2(7)E1	この機能が導入されました。