簡易ネットワーク管理プロトコルの設定

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。最新の機能情報および警告については、使用するプラットフォームおよびソフトウェア リリースのリリース ノートを参照してください。

プラットフォームのサポートおよびシスコ 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 には、 http://www.cisco.com/go/cfnからアクセスします。 Cisco.com のアカウントは必要ありません。

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 *1*：*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	No	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	No	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	No	ユーザ名の照合を使用して認証します。
SNMPv3	authNoPriv	Message Digest 5（MD5）または Secure Hash Algorithm（SHA）	No	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。

モデル	レベル	認証	暗号化	結果
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

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。 エージェントおよび MIB は、controller に常駐します。 controller 上で SNMP を設定するには、マネージャとエージェント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 *2*：*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。[1]
get-bulk-request[2]	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。

[1] 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。

[2] 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 オブジェクトに対するアクセスを認証し、組み込みパスワードとして機能します。 NMS が controller 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 controller 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 controllerがメンバ controllers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 controller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 controller番号（@esN、N はcontroller番号）を追加し、これらのストリングをメンバ controllersに伝播します。

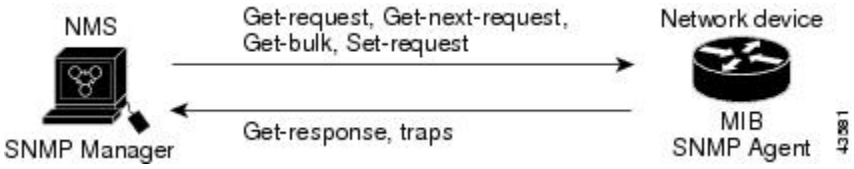
SNMP MIB 変数アクセス

NMS の例として、Cisco Prime Infrastructure ネットワーク管理ソフトウェアがあります。 Cisco Prime Infrastructure 2.0 ソフトウェアは、controller MIB 変数を使用して装置変数を設定し、ネットワーク上の装置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 *1*：*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、controllerから SNMP マネージャに通知を送信できます。 SNMP 通知は、トラップまたはインフォーム要求として送信できます。 コマンド構文では、トラップまたはインフォーム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、インフォーム、またはその両方を表します。 **snmp-server host** コマンドを使用して、トラップまたはインフォームとして SNMP 通知を送信するかどうかを指定します。

（注） SNMPv1 はインフォーム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。 インフォーム要求の場合、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、再びインフォーム要求を送信できます。 再送信できるので、インフォーム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。

インフォームの方がトラップより信頼性が高いのは、controller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、インフォーム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、インフォーム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、トラップにするかインフォームにするかは、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、インフォーム要求を使用してください。 ネットワークまたはcontroller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インターフェイス インデックス（ifIndex）オブジェクト値の生成および割り当てを行います。 controllerの再起動またはcontroller ソフトウェアのアップグレード時に、controller 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、controllerのポート 2 に 10003 という ifIndex 値が割り当てられていると、controllerの再起動後も同じ値が使用されます。

controller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 *3*：*ifIndex* 値

インターフェイス タイプ	**ifIndex 範囲**
SVI[3]	1 ～ 4999
EtherChannel	5000 ～ 5012
ループ バック	5013 ～ 5077
トンネル	5078 ～ 5142
物理（ギガビット イーサネットまたは SFP[4]-モジュール インターフェイス）	10000 ～ 14500
ヌル	14501

3 SVI = スイッチ仮想インターフェイス
4 SFP = 小型フォーム ファクタ

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[5].
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	version 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。

5 これは、controller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

controllerが起動し、controller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 エージェントはイネーブルになります。

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 を設定する場合は、以下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、そのユーザを対応す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、controller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

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、特権 EXEC モードで次の手順を実行します。

はじめる前に

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。

手順の概要

1. **configure terminal**
2. **no snmp-server**
3. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Controller# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**no snmp-server** 例： `Controller(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 3	**end** 例： `Controller(config)# end`	特権 EXEC モードに戻ります。

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、controller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

controller上でコミュニティ ストリングを設定するには、特権 EXEC モードで次の手順を実行します。

手順の概要

1. **configure terminal**
2. **snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**] [*access-list-number*]
3. **access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*]
4. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Controller# 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	**snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**] [*access-list-number*] 例： `Controller(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。

	コマンドまたはアクション	目的
		• （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 3	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] 例： `Controller(config)#` **`access-list 4 deny any`**	（任意）ステップ 2 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 2 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 4	**end** 例： `Controller(config)#` **`end`**	特権 EXEC モードに戻ります。

次に、comaccess ストリングを SNMP に割り当てて読み取り専用アクセスを許可し、IP アクセス リスト 4 がこのコミュニティ ストリングを使用してcontrollerの SNMP エージェントにアクセスできるように指定する例を示します。

```
Controller(config)# snmp-server community comaccess ro 4
```

次の作業

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** コミュニティ ストリング 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。

controller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

controller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。

controller上で SNMP グループとユーザを設定するには、特権 EXEC モードで次の手順を実行します。

手順の概要

1. **configure terminal**
2. **snmp-server engineID** {**local** *engineid-string* | **remote** *ip-address* [**udp-port** *port-number*] *engineid-string*}
3. **snmp-server group** *group-name* {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}} [**read** *readview*] [**write** *writeview*] [**notify** *notifyview*] [**access** *access-list*]
4. **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*]
5. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Controller# 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	**snmp-server engineID** {**local** *engineid-string* | **remote** *ip-address* [**udp-port** *port-number*] *engineid-string*} 例： `Controller(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

	コマンドまたはアクション	目的
		タグラム プロトコル（UDP）ポートを指定します。 デフォルトは162です。
ステップ 3	**snmp-server group** *group-name* {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}} [**read** *readview*] [**write** *writeview*] [**notify** *notifyview*] [**access** *access-list*] 例： `Controller(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 番め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。 • 最も安全な **v3** の場合には、次の認証レベルの 1 つを選択する必要があります。 **auth**：MD5 および SHA によるパケット認証が可能です。 **noauth**：noAuthNoPriv というセキュリティ レベルをイネーブルにします。 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をイネーブルにします（privacy とも呼ばれます）。 （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の名前を表すストリング（64 文字以下）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設定できるビューの名前を表すストリング（64 文字以下）を入力します。 （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の名前を表すストリング（64 文字以下）を入力します。 （任意）**access** *access-list* とともに、アクセス リスト名のストリング（64 文字以下）を入力します。
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** *access-list*] [**auth** {**md5** | **sha**} *auth-password*] } [*priv* {**des** | **3des** | **aes** {**128** | **192** | **256**}} *priv-password*] 例： `Controller(config)#`	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。

	コマンドまたはアクション	目的
	`snmp-server user Pat public v2c`	• **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のみ使用可能です。 • **auth** は認証レベル設定セッションで、HMAC-MD5-96（**md5**）または HMAC-SHA-96（**sha**）認証レベルを使用できます。パスワード ストリング *auth-password*（64 文字以下）が必要です。 **v3** を入力すると、次のキーワードを使用して（64 文字以下）、プライベート（**priv**）暗号化アルゴリズムおよびパスワード ストリング *priv-password* を設定できます。 • **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** は、56 ビット DES アルゴリズムの使用を指定します。 • **3des** は、168 ビット DES アルゴリズムの使用を指定します。 • **aes** は、DES アルゴリズムの使用を指定します。 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 リスト名のストリング（64 文字以下）を入力します。
ステップ **5**	**end** 例： `Controller(config)# end`	特権 EXEC モードに戻ります。

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controller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 Release が稼働している Controllers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。

を **snmp-server host** グローバル コンフィギュレーション コマンドとともに使用して、特定のホストが以下の表に示す通知タイプを受信するようにできます。 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 *4* : デバイスの通知タイプ

通知タイプのキーワード	説明
ブリッジ	STP ブリッジ MIB トラップを生成します。
クラスタ	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。
設定	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。
フラッシュ メモリ	SNMP FLASH 通知を生成します。 controller 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controller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、電源の再投入、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。 controller スタックでは、このトラップはスタックにおけるcontrollerの挿入/取り外しを意味します。
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。
mac-notification	MAC アドレス通知の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 アドバタイズメント、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。

通知タイプのキーワード	説明
pim	プロトコル独立型マルチキャスト（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。
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controllerを設定するには、特権 EXEC モードで次の手順を実行します。

手順の概要

1. **configure terminal**
2. **snmp-server engineID remote** *ip-address engineid-string*
3. **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*] }
4. **snmp-server group** *group-name* {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}} [**read** *readview*] [**write** *writeview*] [**notify** *notifyview*] [**access** *access-list*]
5. **snmp-server host** *host-addr* [**informs** | **traps**] [**version** {**1** | **2c** | **3** {**auth** | **noauth** | **priv**}}] *community-string* [*notification-type*]
6. **snmp-server enable traps** *notification-types*
7. **snmp-server trap-source** *interface-id*
8. **snmp-server queue-length** *length*
9. **snmp-server trap-timeout** *seconds*
10. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Controller# 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	**snmp-server engineID remote** *ip-address engineid-string* 例： `Controller(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 3	**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*] } 例： `Controller(config)# snmp-server user Pat public v2c`	SNMP ユーザを設定し、ステップ 2 で作成したリモート ホストに関連付けます。 （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 4	**snmp-server group** *group-name* {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}} [**read** *readview*] [**write** *writeview*] [**notify** *notifyview*] [**access** *access-list*]	SNMP グループを設定します。

	コマンドまたはアクション	目的
	例： `Controller(config)# snmp-server group public v2c access lmnop`	
ステップ 5	**snmp-server host** *host-addr* [**informs** | **traps**] [**version** {**1** | **2c** | **3** {**auth** | **noauth** | **priv**}}] *community-string* [*notification-type*] 例： `Controller(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 インフォーム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使用可能です。 *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ます。 **version 3** が指定されている場合、SNMPv3 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 6	**snmp-server enable traps** *notification-types* 例： `Controller(config)# snmp-server enable traps snmp`	controller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、**snmp-server enable traps ?** と入力してください。 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。

	コマンドまたはアクション	目的
		（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 7	**snmp-server trap-source** *interface-id* 例： `Controller(config)# snmp-server trap-source GigabitEthernet1/0/1`	（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。
ステップ 8	**snmp-server queue-length** *length* 例： `Controller(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。 指定できる範囲は 1 ～ 1000 です。デフォルトは 10 です。
ステップ 9	**snmp-server trap-timeout** *seconds* 例： `Controller(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 10	**end** 例： `Controller(config)# end`	特権 EXEC モードに戻ります。

次の作業

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 trap** 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）がグローバルで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 コマンドを使用してください。 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、特権 EXEC モードで次の手順を実行します。

手順の概要

1. **configure terminal**
2. **snmp-server contact** *text*
3. **snmp-server location** *text*
4. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Controller# 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	**snmp-server contact** *text* 例： `Controller(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 3	**snmp-server location** *text* 例： `Controller(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 4	**end** 例： `Controller(config)# end`	特権 EXEC モードに戻ります。

SNMP を通して使用する TFTP サーバの制限

SNMP を通してコンフィギュレーション ファイルを保存およびロードするために使用する TFTP（簡易ファイル転送プロトコル）サーバを、アクセス リストに指定されているサーバに限定するには、特権 EXEC モードで次の手順を実行します。

手順の概要

1. **configure terminal**
2. **snmp-server tftp-server-list** *access-list-number*
3. **access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*]
4. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Controller# 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	**snmp-server tftp-server-list** *access-list-number* 例： `Controller(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 3	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] 例： `Controller(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 2 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、controller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。

	コマンドまたはアクション	目的
ステップ 4	**end** 例： `Controller(config)# end`	特権 EXEC モードに戻ります。

SNMP のトラップフラグの設定

手順の概要

1. **configure terminal**
2. **trapflags ap** { **interfaceup | register**}
3. **trapflags client** {**dot11 | excluded**}
4. **trapflags dot11-security** {**ids-sig-attack | wep-decrypt-error**}
5. **trapflags mesh**
6. **trapflags rogueap**
7. **trapflags rrm-params** {**channels | tx-power**}
8. **trapflags rrm-profile** {**coverage | interference | load | noise**}
9. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Controller# 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	**trapflags ap** { **interfaceup	register**} 例： `Controller(config)# trapflags ap interfaceup`	AP 関連トラップの送信をイネーブルにします。 トラップ フラグをディセーブルにするには、このコマンドの **no** 形式を使用します。 • interfaceup：Cisco AP インターフェイス（A または B）が起動したときにトラップをイネーブルにします。 • register：Cisco AP が Cisco controller に登録するときにトラップをイネーブルにします。

	コマンドまたはアクション	目的
ステップ 3	**trapflags client** {**dot11** | **excluded**} 例： `Controller(config)# trapflags client excluded`	クライアント関連 DOT11 トラップの送信をイネーブルにします。トラップ フラグをディセーブルにするには、このコマンドの **no** 形式を使用します。 • dot11：クライアントの DOT11 トラップをイネーブルにします。 • excluded：クライアント用の除外されたトラップをイネーブルにします。
ステップ 4	**trapflags dot11-security** {**ids-sig-attack** | **wep-decrypt-error**} 例： `Controller(config)# trapflags dot11-security wep-decrypt-error`	802.11 セキュリティ関連トラップの送信をイネーブルにします。 トラップ フラグをディセーブルにするには、このコマンドの **no** 形式を使用します。 • ids-sig-attack：IDS シグニチャ攻撃トラップをイネーブルにします。 • wep-decrypt-error：クライアントの WEP 復号化エラーのトラップをイネーブルにします。
ステップ 5	**trapflags mesh** 例： `Controller(config)# trapflags mesh`	メッシュのトラップをイネーブルにします。 トラップフラグをディセーブルにするには、このコマンドの **no** 形式を使用します。
ステップ 6	**trapflags rogueap** 例： `Controller(config)# trapflags rogueap`	不正 AP 検出のトラップをイネーブルにします。 トラップ フラグをディセーブルにするには、このコマンドの **no** 形式を使用します。
ステップ 7	**trapflags rrm-params** {**channels** | **tx-power**} 例： `Controller(config)# trapflags rrm-params tx-power`	RRM-parameter 更新関連トラップの送信をイネーブルにします。 トラップ フラグをディセーブルにするには、このコマンドの **no** 形式を使用します。 • channels：RF マネージャが自動的に Cisco AP インターフェイスのチャネル番号を変更するときにトラップをイネーブルにします。 • tx-power：RF マネージャが自動的に Cisco AP インターフェイスの Tx-Power レベルを変更するときにトラップをイネーブルにします。

	コマンドまたはアクション	目的
ステップ 8	**trapflags rrm-profile** {**coverage** | **interference** | **load** | **noise**} 例： `Controller(config)# trapflags rrm-profile interference`	RRM-Profile 関連トラップの送信をイネーブルにします。 トラップフラグをディセーブルにするには、このコマンドの **no** 形式を使用します。 • coverage：RF マネージャによって保持されるカバレッジ プロファイルでエラーが発生したときにトラップをイネーブルにします。 • interference：RF マネージャによって保持される干渉プロファイルでエラーが発生したときにトラップをイネーブルにします。 • load：RF マネージャによって保持される負荷プロファイルでエラーが発生したときにトラップをイネーブルにします。 • noise：RF マネージャによって保持されるノイズ プロファイルでエラーが発生したときにトラップをイネーブルにします。
ステップ 9	**end** 例： `Controller(config)# end`	特権 EXEC モードに戻ります。

SNMP ワイヤレス トラップ通知のイネーブル化

手順の概要

1. **configure terminal**
2. **snmp-server enable traps wireless** [**AP** | **RRM** | **bsn80211SecurityTrap** | **bsnAPParamUpdate** | **bsnAPProfile** | **bsnAccessPoint** | **bsnMobileStation** | **bsnRogue** | **client** | **mfp** | **rogue**]
3. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Controller# 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	**snmp-server enable traps wireless [AP | RRM | bsn80211SecurityTrap | bsnAPParamUpdate | bsnAPProfile | bsnAccessPoint | bsnMobileStation | bsnRogue | client | mfp | rogue]** 例： `Controller(config)# snmp-server enable traps wireless AP`	SNMP ワイヤレス トラップ通知をイネーブルにします。 • AP：アクセス ポイント トラップをイネーブルにします。 • RRM：RRM トラップをイネーブルにします。 • bsn80211SecurityTrap：セキュリティ関連のトラップをイネーブルにします。 • bsnAPParamUpdate：更新される AP パラメータのトラップをイネーブルにします。 • bsnAPProfile：BSN AP プロファイル トラップをイネーブルにします。 • bsnAccessPoint：BSN アクセス ポイント トラップをイネーブルにします。 • bsnMobileStation：ワイヤレス クライアント トラップを制御します。 • bsnRogue：BSN 不正関連トラップをイネーブルにします。 • client：クライアント トラップをイネーブルにします。 • mfp：MFP トラップをイネーブルにします。 • rogue：不正関連トラップをイネーブルにします。
ステップ 3	**end** 例： `Controller(config)# end`	特権 EXEC モードに戻ります。

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 *5*：***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
	デバイスに設定されているローカル SNMP エンジンおよびすべてのリモートエンジンに関する情報を表示します。
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とき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 での例

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、controller はトラップを送信しません。

```
Controller(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。 controller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。

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

```
Controller(config)# snmp-server community comaccess ro 4
Controller(config)# snmp-server enable traps snmp authentication
Controller(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行めで、controller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 host** コマンドを無効にします。

```
Controller(config)# snmp-server enable traps entity
Controller(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controllerをイネーブルにする例を示します。

```
Controller(config)# snmp-server enable traps
Controller(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ときに **auth**（authNoPriv）認証レベルで情報を送信する例を示します。

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