『SNMP Configuration Guide』

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。

© 2017 Cisco Systems, Inc. All rights reserved.

目 次

最初にお読みください 1

『Configuring SNMP Support』 3

機能情報の確認 3

SNMP サポートの設定について 4

SNMP のコンポーネント 4

SNMP マネージャ 4

SNMP エージェント 4

SNMP MIB 4

SNMP の動作 5

SNMP get 5

SNMP SET 5

SNMP 通知 5

トラップとインフォーム 6

MIB および RFC 8

SNMP のバージョン 8

インターフェイス登録の詳細情報 10

インターフェイス インデックス 11

インターフェイス エイリアス 11

Interface Name 11

VPN のための SNMP サポート 12

インターフェイス インデックスのパーシステンス 12

インターフェイス インデックス パーシステンスの利点 13

『Event MIB』 13

Event 13

オブジェクト リスト 14

Trigger 14

トリガー テスト 14

式 MIB 14
絶対値サンプリング 15
デルタ サンプリング 15
変更サンプリング 15
SNMP 通知ロギング 15
SNMP サポートの設定方法 16
システム情報の設定 16
SNMP バージョン 1 および 2 の設定 18
前提条件 18
SNMP ビュー レコードの作成または変更 18
SNMP コミュニティの作成または変更のアクセス コントロール 19
SNMP トラップ操作の受信者の設定 21
SNMP バージョン 3 の設定 23
SNMP サーバ グループ名の指定 23
SNMP サーバ ユーザの設定 25
デバイスを SNMP マネージャとして設定 27
SNMP マネージャのイネーブル化 30
SNMP エージェント シャットダウン メカニズムのイネーブル化 32
SNMP エージェント パケットの最大サイズの定義 33
SNMP を介して使用される TFTP サーバ数の制限 34
トラブルシューティングのヒント 35
SNMP エージェントのディセーブル化 35
SNMP 通知の設定 36
SNMP 通知を送信するようにデバイスを設定する 37
通知操作値の変更 39
個々の RFC 1157 SNMP トラップの制御 40
SNMP 通知ログ オプションの設定 43
インターフェイス インデックス表示の設定とインターフェイス インデックスおよびロング ネーム サポート 44
インターフェイス インデックス パーシステンスの設定 48
ifIndex パーシステンスのグローバルなイネーブル化とディセーブル化 48

特定のインスタンスでの ifIndex パーシステンスのイネーブル化およびディセーブル化 49
VPN のための SNMP サポートの設定 51
SNMP を使用するイベント MIB の設定 53
トリガー テーブルでのトリガーの設定 53
イベント テーブルでのイベントの作成 55
トリガー テーブルでのトリガーのしきい値の設定とアクティブ化 56
トリガーのアクティブ化 57
イベント MIB のモニタリングおよびメンテナンス 57
コマンドライン インターフェイスを使用したイベント MIB の設定 58
スカラ変数の設定 58
イベント MIB オブジェクト リストの設定 59
イベントの設定 60
イベント アクションの設定 62
アクション通知の設定 62
アクション セットの設定 63
イベント トリガーの設定 64
存在トリガー テストの設定 66
ブール値トリガー テストの設定 68
しきい値トリガー テストの設定 69
SNMP を使用する式 MIB の設定 71
CLI を使用した式 MIB の設定 74
式 MIB のスカラ オブジェクトの設定 74
式の設定 76
SNMP サポートの設定例 79
例：SNMPv1、SNMPv2c、および SNMPv3 の設定 79
例：ifAlias ロング ネーム サポートの設定 81
例：VPN のための SNMP サポートの設定 82
例：イベント MIB の設定 82
例：式 MIB の設定 84
その他の参考資料 84
SNMP サポートの設定に関する機能情報 87

用語集 92
SNMP Support over VPNs：コンテキストベース アクセス コントロール 93
機能情報の確認 93
SNMP Support over VPNs の制限事項：コンテキストベース アクセス コントロール 94
SNMP Support over VPNs に関する情報：コンテキストベース アクセス コントロール 94
SNMP のバージョンとセキュリティ 94
SNMPv1 または SNMPv2 セキュリティ 94
SNMP Notification Support over VPNs 95
VPN 対応 SNMP 96
VPN ルート識別子 96
SNMP コンテキスト 96
SNMP Support over VPNs の設定方法--コンテキストベース アクセス コントロール 97
SNMP コンテキストの設定および SNMP コンテキストと VPN の関連付け 97
SNMP サポートの設定および SNMP コンテキストの関連付け 100
SNMP Support over VPNs の設定例：コンテキスト ベース アクセス コントロール 102
例：コンテキストベース アクセス コントロールの設定 102
その他の参考資料 103
SNMP Support over VPNs に関する機能情報：コンテキストベース アクセス コントロール 106
SNMP バージョン 3 のための AES および 3-DES 暗号化サポート 107
機能情報の確認 107
SNMP バージョン 3 の AES および 3 DES 暗号化サポートの前提条件 108
SNMP バージョン 3 のための AES および 3 DES 暗号化サポートに関する情報 108
AES および 3-DES 暗号化サポートの概要 109
暗号キーのサポート 110
MIB サポート 110
SNMP バージョン 3 のための AES および 3 DES 暗号化サポートの設定方法 110
SNMP グループへの新しいユーザの追加 110
SNMP ユーザの設定の確認 111
その他の参考資料 112
SNMP バージョン 3 の AES および 3 DES 暗号化サポートの機能情報 113

VLAN サブインターフェイスに対する簡易ネットワーク管理プロトコル（SNMP）サポート 115

機能情報の確認 **115**

VLAN サブインターフェイスに対する SNMP サポートについて **116**

利点 **116**

サポートされるプラットフォーム **116**

VLAN サブインターフェイスに対する SNMP サポートの仕組み **116**

VLAN サブインターフェイスでの SNMP エージェントのイネーブル化 **116**

VLAN サブインターフェイスでの SNMP サポートの設定例 **118**

例：VLAN サブインターフェイスでの SNMP エージェントのイネーブル化 **118**

その他の参考資料 **119**

VLAN サブインターフェイスでの SNMP サポートの機能情報 **120**

メモリ プール：SNMP 通知のサポート 121

機能情報の確認 **121**

メモリ プールの前提条件：SNMP 通知のサポート **122**

メモリ プールの制限事項：SNMP 通知のサポート **122**

メモリ プールに関する情報：SNMP 通知のサポート **122**

メモリ プールをイネーブルにする方法：SNMP 通知のサポート **123**

メモリ プールの設定例：SNMP 通知のサポート **124**

メモリ プールのイネーブル化：SNMP 通知サポートの例 **124**

その他の参考資料 **124**

メモリ プールの機能情報：SNMP 通知のサポート **127**

定期的な管理情報ベース（MIB）データの収集および転送のメカニズム 129

機能情報の確認 **129**

定期的な MIB データの収集および転送のメカニズムの前提条件 **130**

定期的な MIB データの収集および転送のメカニズムの制約事項 **130**

定期的な MIB データの収集および転送のメカニズムについて **130**

SNMP のオブジェクトとインスタンス **130**

バルク統計情報オブジェクト リスト **131**

バルク統計情報スキーマ **131**

バルク統計情報転送オプション **131**

定期的な MIB データの収集および転送のメカニズムの利点 **132**

定期的な MIB データの収集および転送のメカニズムの設定方法 132
バルク統計情報オブジェクト リストの設定 132
バルク統計情報スキーマの設定 134
バルク統計情報転送オプションの設定 136
トラブルシューティングのヒント 140
バルク統計情報収集のモニタリングのイネーブル化 141
定期的な MIB データの収集および転送のメカニズムのモニタリングとトラブル シューティング 142
定期的な MIB データの収集および転送のメカニズムの設定例 144
定期的な MIB データの収集および転送のメカニズムの設定の例 144
転送パラメータ 144
ポーリングの要件 145
オブジェクト リストの設定 145
スキーマ定義の設定 146
転送パラメータの設定 146
ステータスの表示 146
バルク統計情報出力ファイル 147
その他の参考資料 148
定期的な MIB データの収集および転送のメカニズムの機能情報 149

第 1 章

最初にお読みください

Cisco IOS XE 16 についての重要事項

Cisco IOS XE Release 3.7.0E（Catalyst スイッチ）と Cisco IOS XE Release 3.17S（アクセスおよびエッジ ルーティング）という有効な 2 つのリリースが統合され、スイッチングおよびルーティング ポートフォリオ内のアクセスおよびエッジ製品を幅広く網羅する 1 つの統合リリース バージョン（Cisco IOS XE 16）へと進化しました。

（注） 機能が導入されると、技術構成ガイドの [Feature Information] テーブルで通知されます。その機能に対応している他のプラットフォームについては、通知される場合と通知されない場合があります。特定の機能がプラットフォームでサポートされているかどうかを確認するには、製品のランディング ページに表示される技術構成ガイドをご覧ください。製品のランディング ページに技術構成ガイドが表示された場合、そのプラットフォームでは機能がサポートされています。

第 2 章

『Configuring SNMP Support』

簡易ネットワーク管理プロトコル（SNMP）は、アプリケーション層プロトコルであり、SNMP マネージャと SNMP エージェントとの通信に使用されるメッセージ フォーマットを提供します。SNMP はネットワーク デバイスのモニタリングや管理に使用される標準化されたフレームワークと共通言語を提供します。

このマニュアルでは、シスコ デバイスで SNMP エージェントをイネーブルにする方法、エージェントからの SNMP 通知の送信を制御する方法について説明します。SNMP 管理システムの使用の詳細については、ご使用のネットワーク管理システム（NMS）アプリケーションのマニュアルを参照してください。

- 機能情報の確認，3 ページ
- SNMP サポートの設定について，4 ページ
- SNMP サポートの設定方法，16 ページ
- SNMP サポートの設定例，79 ページ
- その他の参考資料，84 ページ
- SNMP サポートの設定に関する機能情報，87 ページ
- 用語集，92 ページ

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、Bug Search Tool およびご使用の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。このモジュールで説明される機能に関する情報、および各機能がサポートされるリリースの一覧については、機能情報の表を参照してください。

プラットフォームのサポートおよびシスコ 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。

SNMP サポートの設定について

SNMP のコンポーネント

簡易ネットワーク管理プロトコル（SNMP）は、SNMP マネージャとエージェント間の通信用メッセージ 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 はネットワーク デバイスのモニタリングや管理に使用される標準化されたフレームワークと共通言語を提供します。

SNMP フレームワークには、次のコンポーネントがあります。これらについては次の各項で説明します。

SNMP マネージャ

Simple Network Management Protocol（SNMP）マネージャは、SNMP を使用するネットワーク ホストのアクティビティを制御およびモニタするシステムです。最も一般的な管理システムは、ネットワーク管理システム（NMS）です。NMS という用語は、ネットワーク管理に使用する専用デバイスを意味する場合と、このようなデバイス上で使用するアプリケーションを意味する場合があります。SNMP では、単純なコマンドライン インターフェイス アプリケーションから CiscoWorks2000 製品などの GUI を使用するアプリケーションまで、いくつかのネットワーク管理アプリケーションを使用できます。

SNMP エージェント

Simple Network Management Protocol（SNMP）エージェントは、デバイスに対してデータを保守し、このデータを必要に応じて管理システムにレポートする管理対象デバイス内のソフトウェア コンポーネントです。このエージェントは、ルーティング デバイス（ルータ、アクセス サーバ、スイッチなど）内に存在します。Cisco ルーティング デバイス上で SNMP エージェントをイネーブルにするには、マネージャとエージェントの関係を定義する必要があります。

（注） 多くのシスコ デバイスが SNMP エージェントになるように設定できますが、この方法は推奨されません。エージェントで SNMP プロセスを制御するために必要なコマンドは、Cisco コマンドライン インターフェイスから追加の設定なしで使用できます。

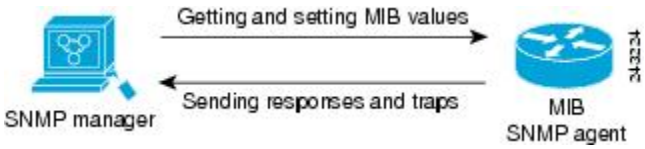
SNMP MIB

SNMP エージェントには、SNMP マネージャが Get 操作や Set 操作を通じて要求したり変更したりできる MIB 変数が含まれています。マネージャでは、エージェントからの値の取得またはエージェントへの値の保存が可能です。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

に関する情報のリポジトリである SNMP MIB から値を収集します。エージェントは、マネージャのデータ取得要求やデータ設定要求にも応答できます。

以下の図は、SNMP マネージャとエージェント間の通信を示します。マネージャは、SNMP MIB 値を取得および設定する要求をエージェントに送信します。エージェントは要求に応答します。この相互作用に関係なく、エージェントは要求されていない通知（トラップまたはインフォーム）をマネージャに送信して、マネージャにネットワークの状態を通知できます。

図 *1*：*SNMP* エージェントと *SMNP* マネージャの間の通信



SNMP の動作

Simple Network Management Protocol（SNMP）アプリケーションは、次の操作を実行して、データの取得、SNMP オブジェクト変数の変更、および通知の送信を行います。

SNMP get

Simple Network Management Protocol（SNMP）の GET 動作が、Network Management Server（NMS）によって実行され、SNMP オブジェクト変数が取得されます。GET 動作には、次の 3 つのタイプがあります。

- GET：SNMP エージェントから、オブジェクト インスタンスそのものを取得します。
- GETNEXT：その次のオブジェクト変数を取得します。辞書順で、指定された変数の次になる値です。
- GETBULK：GETNEXT 操作を繰り返さなくても、大量のオブジェクト変数データを取得できます。

SNMP SET

Simple Network Management Protocol（SNMP）の SET 動作が、Network Management Server（NMS）によって実行され、オブジェクト変数の値が変更されます。

SNMP 通知

Simple Network Management Protocol（SNMP）の重要な機能の 1 つは、要求されていない通知を SNMP エージェントから作成できるということです。

トラップとインフォーム

要求されていない（非同期の）通知は、トラップ要求またはインフォーム要求（informs）として生成できます。トラップは、Simple Network Management Protocol（SNMP）マネージャにネットワークの状態を通知するメッセージです。インフォームは、SNMP マネージャからの受信確認の要求を含むトラップです。通知は、不正なユーザ認証、再起動、接続の切断、隣接デバイスとの接続の切断、その他の重要なイベントを表示します。

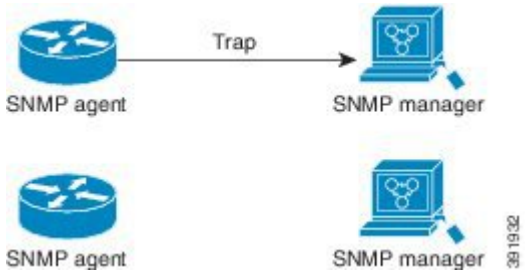
トラップの信頼性はインフォームより低くなります。受信側はトラップを受信しても確認応答を送信しないからです。送信者は、トラップが受信されたかどうかを知ることができません。インフォームを受信したSNMPマネージャは、SNMP応答プロトコルデータユニット（PDU）を使用してメッセージに確認応答します。送信者は、応答を受信できない場合には、再びインフォームを送信します。このため、インフォームの方が目的の宛先に到達する確実性が高くなります。

トラップは信頼性が劣りますが、インフォームよりも推奨されます。これは、インフォームはデバイスおよびネットワーク内のリソース消費が多いからです。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、メモリ内に保持する必要があります。またトラップが一度だけ送信されるのに対し、インフォームは数回再試行されることがあります。再試行によってトラフィックが増え、ネットワークのオーバーヘッドが大きくなる原因になります。トラップとインフォームの使用では、信頼性とリソースの兼ね合いを考える必要があります。SNMP マネージャがすべての通知を受信することが重要な場合は、インフォームを使用してください。ただし、トラフィック量またはメモリ使用量が問題であり、通知の受信が必要ない場合は、トラップを使用してください。

下図に、トラップとインフォームの違いを示します。

下図は、エージェントが SNMP マネージャへ正常にトラップを送信した場合を示します。マネージャはトラップを受信しても、確認応答を送信しません。エージェントには、トラップが宛先に到達したことを知る方法がありません。

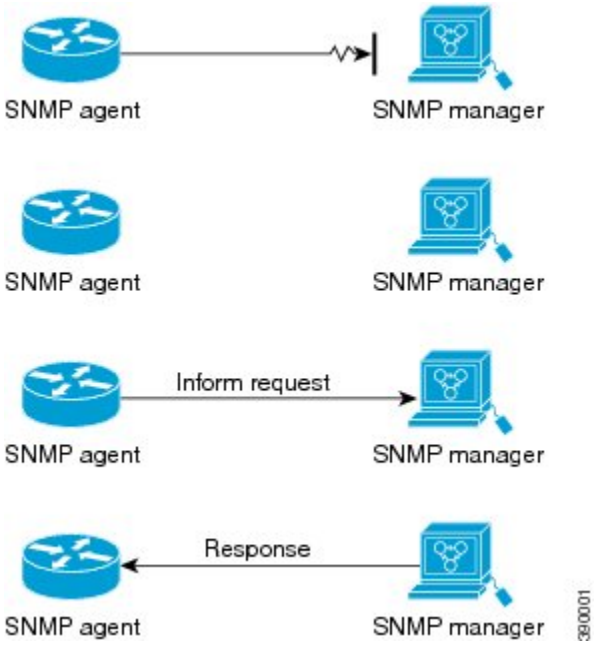
図 *2*：*SNMP* マネージャに正常に送信されたトラップ



下図では、エージェントはマネージャへ正常にインフォームを送信しています。マネージャがインフォームを受信すると、応答がエージェントに送信されます。これにより、エージェントはイ

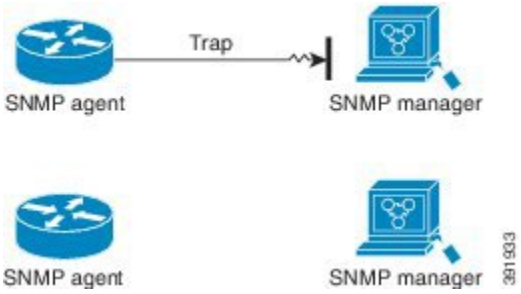
ンフォームが宛先に到達したことがわかります。この例では、上図に示すインタラクション内で 2 倍のトラフィックが生成されていることに注意してください。

図 *3*：*SNMP* マネージャに正常に送信されたインフォーム要求



下図は、マネージャが受信しないトラップをマネージャに送信するエージェントを示します。エージェントには、トラップが宛先に到達しなかったことを知る方法がありません。トラップが再送信されないため、マネージャはトラップを受信しません。

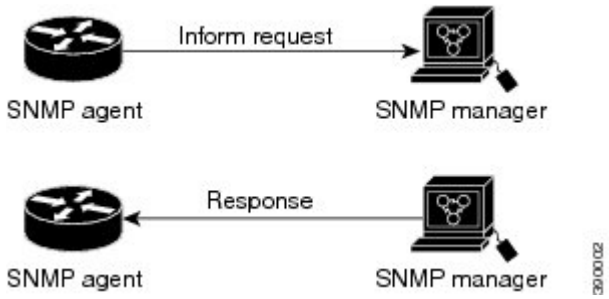
図 *4*：*SNMP* マネージャに正常に送信されなかったトラップ



下図は、マネージャに到達しないインフォームをマネージャに送信するエージェントを示します。マネージャはインフォームを受信しなかったため、応答を送信しません。一定時間が経過すると、エージェントがインフォームを再送信します。マネージャは、2 番目の送信からインフォームを

受信して応答します。この例では、上図に示すシナリオよりも多くのトラフィックが生成されますが、通知は SNMP マネージャに到達します。

図 *5*：*SNMP* マネージャに正常に送信されなかったインフォーム



MIB および RFC

MIB モジュールは通常、国際標準団体である IETF に提出された RFC 文書で規定されます。RFC は、インターネット学会およびインターネット コミュニティ全体による意向に基づいて、通常は推奨されるインターネット標準を確立することを目的として、個人または団体によって記述されます。RFC ステータスを付与される前に、推奨事項が Internet Draft（I-D）文書として発行されます。推奨標準となった RFC には標準文書（STD）のラベルも付けられます。標準の手続きと IETF の活動については、Internet Society の Web サイト（http://www.isoc.org）を参照してください。シスコのマニュアルで参照されているすべての RFC、I-D、および STD の全文は、IETF の Web サイト（http://www.ietf.org）で参照できます。

シスコの SNMP 実装は、RFC 1213 で記述される MIB II 変数の定義および RFC 1215 で記述される Simple Network Management Protocol（SNMP）トラップの定義を使用しています。

Cisco はすべてのシステムに専用の拡張 MIB を提供します。Cisco エンタープライズ MIB は、マニュアル内で特に言及しない限り、関連 RFC で記述されるガイドラインに準拠しています。Cisco.com にある Cisco MIB の Web サイトで、MIB モジュール定義ファイルおよび各 Cisco プラットフォームでサポートされる MIB のリストを参照できます。

SNMP のバージョン

Cisco IOS ソフトウェアでは、次のバージョンの SNMP がサポートされています。

- SNMPv1：RFC 1157 で規定された、完全なインターネット標準のシンプル ネットワーク管理プロトコル（RFC 1157 は、RFC 1067 および RFC 1098 として公開されたそれより古いバージョンを置き換えるものです）。コミュニティ ストリングに基づいてセキュリティを実現します。
- SNMPv2c：SNMPv2 のためのコミュニティ ストリング ベースの管理フレームワーク。SNMPv2c（「c」は「community」を示す）は RFC 1901、RFC 1905、および RFC 1906 で規定されている実験的なインターネット プロトコルです。SNMPv2c は、SNMPv2p（SNMPv2 Classic）の

プロトコル操作とデータ タイプが更新されたもので、SNMPv1 のコミュニティベースのセキュリティ モデルを使用します。

- SNMPv3：SNMP バージョン 3。SNMPv3 は、RFC 3413 ～ 3415 で規定される、相互運用可能な標準ベースのプロトコルです。SNMPv3 は、ネットワーク経由のパケットの認証と暗号化によって、デバイスへのセキュア アクセスを実現します。

SNMPv3 が提供するセキュリティ機能は、次のとおりです。

- メッセージの完全性：パケットが伝送中に改ざんされていないことを保証します。
- 認証：有効な送信元からのメッセージであることを判別します。
- 暗号化：パケットの内容をスクランブル化することにより、不正な送信元に学習されないようにします。

SNMPv1 および SNMPv2c はどちらも、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 MIB にアクセスできる SNMP マネージャのコミュニティは、コミュニティ ストリングによって定義されます。

SNMPv2C のサポートには、バルク検索メカニズムと、詳細なエラー メッセージを管理ステーションに報告する機能が含まれます。一括取得メカニズムは、テーブルおよび大量の情報の取得をサポートして、必要なラウンド トリップの回数を最小化します。SNMPv2C の改良されたエラー処理のサポートには多様なエラー状態を区別する拡張型エラー コードがあります。これらの状態は、SNMPv1 では 1 つのエラー コードで報告されます。no such object、no such instance、および end of MIB view の 3 つのタイプの例外もレポートされます。

SNMPv3 は、ユーザおよびユーザが属しているグループに対して認証方式が設定されるセキュリティ モデルです。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 モデルとセキュリティ レベルの組み合わせにより、SNMP パケット処理中に採用されるセキュリティ メカニズムが決まります。

SNMPv1、SNMPv2c、および SNMPv3 の 3 つのセキュリティ モデルを使用できます。次の表に、セキュリティ モデルとレベルの組み合わせおよびその意味を示します。

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
v1	noAuthNoPriv	Community String	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。
v2c	noAuthNoPriv	Community String	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。

モデル	レベル	認証	暗号化	結果
v3	noAuthNoPriv	Username	No	ユーザ名の照合を使用して認証します。
v3	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。
v3	authPriv	MD5 または SHA	データ暗号規格（DES）	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。DES 56 ビット暗号化を提供し、CBC-DES（DES-56）標準に基づいて認証します。

（注） SNMPv2p（SNMPv2 Classic）は、Cisco IOS Release 11.2 以降ではサポートされません。SNMPv2c は、SNMPv2p のパリティ 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 ベースの管理フレームワークに置き換えます。SNMPv2c でも、SNMPv2p の一括取得とエラー処理の機能は残されています。

管理ステーションでサポートされているバージョンの SNMP を使用するように 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ます。Cisco IOS ソフトウェアを設定して、SNMPv1 プロトコルを使用した管理ステーションとの通信、SNMPv2c プロトコルを使用した管理ステーションとの通信、および SNMPv3 を使用したその他の管理ステーションとの通信に対応することができます。

SNMPv3 は、RFC 1901 ～ 1908、2104、2206、2213、2214、および 2271 ～ 2275 をサポートします。SNMPv3 の詳細については、RFC 2570『*Introduction to Version 3 of the Internet-standard Network Management Framework*』を参照してください（これは標準文書ではありません）。

インターフェイス登録の詳細情報

SNMP のためのインターフェイス インデックス表示の機能は、SNMP の上級ユーザがインターフェイスの登録に関する情報を管理対象エージェント上で直接表示できるようにするために、新しい

コマンドとコマンドの変更を導入します。外部 NMS を使用しなくてもエージェントから MIB 情報を表示できます。

（注） このマニュアルの目的では、エージェントは、Cisco ソフトウェアを実行しているルーティング デバイスです。

この機能は、インターフェイス MIB 内の 3 つのオブジェクト、ifIndex、ifAlias、および ifName を対象とします。これらのオブジェクトの完全な定義については、Cisco SNMPv2 MIB の Web サイトから入手できる IF-MIB.my ファイルを参照してください。

インターフェイス インデックス

ifIndex オブジェクト（ifEntry 1）はインターフェイス インデックスと呼ばれます。インターフェイス インデックスは、管理対象デバイス上の各インターフェイスまたはサブインターフェイスを識別する、0 よりも大きい一意の値です。この値はインターフェイス インデックスの識別番号となります。

CLI コマンド **show snmp mib ifmib ifindex** では、インターフェイスおよびサブインターフェイスに割り当てられた SNMP インターフェイス インデックス識別番号を表示できます。NMS は必要ではありません。

インターフェイス エイリアス

ifAlias オブジェクト（ifXEntry 18）はインターフェイス エイリアスと呼ばれます。インターフェイス エイリアスは、SNMP ネットワーク管理に使用されるインターフェイスに対してユーザが指定した説明です。ifAlias は、インターフェイス グループ MIB（IF-MIB）内のオブジェクトです。このオブジェクトは、ネットワーク マネージャがインターフェイスの「名前」に設定できます。インターフェイスまたはサブインターフェイスの ifAlias 値は、インターフェイス コンフィギュレーション モードまたはサブインターフェイス コンフィギュレーション モードで **description** コマンドを使用するか、または NMS から set 操作を使用して設定できます。以前は、サブインターフェイスの ifAlias の説明は 64 文字に制限されていました（OLD-CISCO-INTERFACES-MIB では locIfDescr MIB 変数に 255 文字まで入力できますが、この MIB はサブインターフェイスをサポートしません）。新しい CLI コマンド、**snmp ifmib ifalias long** は、256 文字までの IfAlias の説明を処理するようにシステムを設定できます。ifAlias の説明は、CLI コマンド **show interfaces** の出力に表示されます。

Interface Name

ifName オブジェクト（ifXEntry 1）は、インターフェイスのテキストでの名前です。ifName オブジェクトの目的は、与えられたインターフェイスの CLI での表現を相互参照することです。このオブジェクトの値は、ローカル デバイスによって割り当てられたインターフェイスの名前で、通常は CLI コマンドでの使用に適しています。ローカル名がない場合や、そうでなくてもこのオブジェクトが適用されない場合、このオブジェクトには長さが 0 の文字列が格納されます。この機

能で導入されたコマンドは ifName オブジェクトに反映されませんが、ここでは ifIndex オブジェクトおよび ifAlias オブジェクトとの関係を示すために説明します。

show snmp mib コマンドでは、Cisco デバイス上の MIB ですべてのオブジェクトが表示されます（mibwalk と同様）。MIB ツリー内のオブジェクトは、文字の順序を使用してソートされます。つまり、オブジェクト ID が番号順に保存されます。NMS の GetNext 操作は、オブジェクト ID（OID）または部分 OID を入力として受け取り、ツリーの文字の順序に基づいて MIB ツリー内から取得した次のオブジェクトを返すため、この操作を使用する場合には文字の順序が重要です。

VPN のための SNMP サポート

VPN 機能の SNMP サポートでは、SNMP トラップおよびインフォームを VPN ルーティング/転送（VRF）テーブルを使用して送受信できます。この機能により、特に、個々の VPN 固有の SNMP トラップおよび SNMP インフォームを送受信するためのサポートが Cisco IOS ソフトウェアに追加されます。

VPNは、プライベートネットワークと同じ使用ガイドラインを持つ共有システム上での接続性の高い転送を提供するネットワークです。VPN は IP でのインターネット、フレーム リレー、または ATM ネットワーク上で構築できます。

VRF では VPN ごとにルーティング データを保存します。これはネットワーク アクセス サーバ（NAS）に接続された顧客のサイトの VPN メンバーシップを定義します。VRF は、IP ルーティング テーブル、派生 Cisco Express Forwarding テーブル、およびルーティング テーブルに含まれる情報を制御するガイドラインとルーティング プロトコルのパラメータで構成されます。

VPNのためのSNMPサポートの機能は、ユーザがSNMPエージェントとマネージャを特定のVRFと関連付けることを可能にするコンフィギュレーション コマンドを提供します。指定されたVRFを使用して、SNMP トラップと SNMP インフォーム、およびエージェントとマネージャ間の応答の送信が実行されます。VRFが指定されなかった場合は、そのVPNのデフォルトのルーティングテーブルが使用されます。

VPN のサポートにより、特定の VPN セットからの SNMP 要求のみを受け取るように SNMP エージェントを設定できます。この設定では、サービス プロバイダーが顧客に、すべてのユーザの VPN デバイスを管理できるネットワーク管理サービスを提供できます。

インターフェイス インデックスのパーシステンス

SNMPベースのネットワーク管理アプリケーションで最も一般的に使用されるIDの1つは、インターフェイスインデックス（IfIndex）値です。IfIndexは、物理または論理インターフェイスに関連付けられる一意のIDで、ほとんどのソフトウェアに関していえば、ifIndexはインターフェイスの名前です。

特定のifIndex値とそのインターフェイスの間の通がリブート後も維持されるという要件は、関連RFCに存在しませんが、デバイスのインベントリ、請求、障害検出などのアプリケーションでは、この通信の維持への依存度が高まっています。

この機能では、リブート後も維持される ifIndex 値のサポートが追加され、以前は一貫したインターフェイス ID を保つために必要だった回避策が不要になります。

現在、一定間隔でデバイスにポーリングすることによって、インターフェイスと ifIndex の関連付けが可能ですが、このインターフェイスを常にポーリングすることは現実的ではありません。このデータが定常的に関連付けられていないと、ポーリング間のリブートやデバイスへの新しいカードの挿入でデータが正しくなくなる可能性があります。したがって、ifIndex パーシステンスは、データ整合性を保証するための唯一の方法です。

ifIndex パーシステンスでは、ifDescr オブジェクト値と ifIndex オブジェクト値（IF-MIB から生成される）のマッピングが、リブート後も維持されます。

インターフェイス インデックス パーシステンスの利点

ネットワーク管理のためのインターフェイスとトラフィック ターゲットのアソシエーション

インターフェイス インデックス パーシステンス機能は、トラフィック フローおよび SNMP 統計情報の入力インターフェイスと出力インターフェイスを一意に識別することにより、ネットワーク管理データの収集と処理を行う際の精度を高めます。各インターフェイスを既知のエンティティ（ISP の加入者など）に関連付けることにより、ネットワーク管理データがより効果的に使用できるようになります。

メディエーション、障害検出、および請求の精度

ネットワーク データは、使用状況に基づいた課金、ネットワーク プランニング、ポリシーの強制、傾向分析などのために、世界中でますます使用されるようになっています。ifIndex 情報は、トラフィック フローおよび SNMP 統計情報の入力インターフェイスと出力インターフェイスを識別するために使用されます。各インターフェイスを既知のエンティティ（加入者など）に確実に関連付けることができないと、データの有効性がなくなります。

『Event MIB』

イベント MIB では SNMP を使用してローカル システムまたはリモート システム上の MIB オブジェクトをモニタし、トリガー条件が満たされると単純なアクションを開始する機能が提供されます。たとえば、SNMP トラップはオブジェクトが変更されると生成できます。イベントを通じて通知がトリガーされる場合は、NMS が定常的に管理対象デバイスをポーリングして変更を追跡する必要はありません。

指定された条件が満たされた場合のみに SNMP 通知が実行されるようにすると、イベント MIB が影響を受けるデバイス上の負荷を軽減し、ネットワーク管理ソリューションの拡張性が向上します。

イベント MIB はイベント、イベントに設定されたオブジェクト リスト、イベント アクション、トリガー、トリガー テストに基づいて動作します。

Event

イベント テーブルでは、イベントがトリガーされたときに実行するアクティビティを定義します。実行されるアクティビティには、通知の送信、MIB オブジェクトの送信などがあります。イ

ベントテーブルは、イベントアクションに従って設定されたその他のオブジェクトのための補助テーブルも持ちます。イベント アクションとして通知が設定されている場合は、そのイベントに設定されているオブジェクトに変更が加えられるたびに通知が送信されます。

オブジェクト リスト

オブジェクト テーブルには、トリガー、トリガー テスト タイプ、または通知を送信するイベントに基づいて通知に追加できるオブジェクトをリストします。イベント MIB では、ワイルドカードを使用できます。これにより、オブジェクトの複数のインスタンスをモニタできます。オブジェクト ID のグループの指定では、ワイルドカード オプションを使用できます。

Trigger

トリガーテーブルでは、イベントをトリガーするための条件を定義します。トリガーテーブルにはモニタ対象のオブジェクトが表示され、各トリガーがイベントに関連付けられます。トリガーがアクティブ化されると、イベントが発生します。トリガーを作成するには、イベント MIB の mteTriggerTable 内でトリガー エントリを設定します。このトリガー エントリでは、モニタ対象とするオブジェクトのオブジェクト ID を指定します。各トリガーは、単一のオブジェクト、またはワイルドカード（*）によって指定したオブジェクトのグループをモニタするように設定されます。イベント MIB プロセスでは、モニタ対象オブジェクトの状態が、指定された間隔でチェックされます。

トリガー テスト

トリガーテーブルには、トリガーに対して実行されるテストタイプに基づいて設定された追加オブジェクトの補足テーブルがあります。存在、しきい値、ブール値などの各トリガーエントリ タイプに対して、対応するテーブル（existence テーブル、threshold テーブル、および Boolean テーブル）に、テストの実行に必要な情報が格納されています。イベント MIB では、存在、しきい値、およびブール値の 3 つのトリガー タイプに基づいてイベント トリガーを設定できます。オブジェクトに対して指定したテストで *true* の値が返された場合、トリガーがアクティブになっています。イベント MIB は、トリガーがアクティブになったら関連のあるホストに通知を送信するように設定できます。

式 MIB

式 MIB では、オブジェクトの組み合わせに基づいて式を作成できます。サンプリング方式に従って式が評価されます。式 MIB では、次のタイプのオブジェクト サンプリングがサポートされます。

- Absolute
- 差分
- 変更

式の中にデルタ値や変更値がない場合は、要求元が式の値を読み取ろうとしたときに式が評価されます。この場合、すべての要求元が新しく計算された値を取得します。

デルタ値または変更値を持つ式の場合は、サンプリングのたびに評価が実行されます。この場合、要求元が最後のサンプル期間の値を取得します。

絶対値サンプリング

絶対値サンプリングは、サンプリング中に MIB オブジェクトの値を使用します。

デルタ サンプリング

デルタ サンプリングは、1 つのサンプルから次のサンプルまでのデルタ（差分）に基づいて識別されるカウンタを持つ式に使用されます。デルタ サンプリングは前回のサンプルの値を使用するため、このサンプリングには、アプリケーションが連続的にサンプリングを行う必要があります。

変更サンプリング

変更サンプリングでは、最後のサンプル以降のオブジェクトの変更された値を使用します。

SNMP 通知ロギング

SNMP をサポートするシステムは、多くの場合、通知情報を記録するためのメカニズムを必要とします。このメカニズムで、再送信の制限を超えた通知が失われないようにします。通知ログMIBは、ローカルロギング機能の形で、その他のMIBのための共通インフラストラクチャを提供します。SNMP 通知ロギング機能は、コマンドラインで通知ログのサイズ変更、グローバルな ageout 値の設定、およびロギング サマリーの表示を行うための Cisco コマンドライン インターフェイス コマンドを追加します。通知ログ MIB では、通知の追跡が向上し、すべての MIB の追跡のための一元的な場所が提供されます。

authenticationFailure、linkUp、linkDown、warmStart、coldStart の各トラップまたはインフォームは、グローバルにイネーブルまたはディセーブルにできます（これらのトラップはRFC 1157 で定義されている「generic trap」の一部です）。linkUp および linkDown 通知は特定のインターフェイスではデフォルトでイネーブルになっていますが、グローバルにイネーブルにされない限り送信されないことに注意してください。

（注） 通知ログ MIB では、デフォルト ログのみで通知のロギングがサポートされます。

SNMP サポートの設定方法

SNMP をイネーブルにするために使用する特定のコマンドはありません。最初に **snmp-server** コマンドを入力すると、SNMP のサポートされるバージョンがイネーブルになります。その他の設定はすべて任意です。

システム情報の設定

コンフィギュレーション ファイルからアクセスできるように、SNMP エージェントのシステム接点、場所、およびシリアル番号などの説明を設定できます。ここで説明する設定手順は任意ですが、これらの基本情報は設定をトラブルシューティングする際に役立つため、設定しておくことを推奨します。さらに、初めて **snmp-server** コマンドを実行すると、デバイス上の SNMP がイネーブルになります。

この手順は、必要に応じて実行してください。

手順の概要

1. イネーブル化
2. **configureterminal**
3. **snmp-servercontact***text*
4. **snmp-serverlocation***text*
5. **snmp-serverchassis-id***number*
6. end
7. **showsnmpcontact**
8. **showsnmplocation**
9. **showsnmpchassis**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	イネーブル化 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	**snmp-servercontact***text* 例： `Device(config)# snmp-server contact NameOne`	システムの連絡先文字列を設定します。
ステップ 4	**snmp-serverlocation***text* 例： `Device(config)# snmp-server location LocationOne`	システムの場所を表す文字列を設定します。
ステップ 5	**snmp-serverchassis-id***number* 例： `Device(config)# snmp-server chassis-id 015A619T`	システムのシリアル番号を設定します。
ステップ 6	end 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了します。
ステップ 7	**showsnmpcontact** 例： `Device# show snmp contact`	（任意）システムに設定されている連絡先文字列を表示します。
ステップ 8	**showsnmplocation** 例： `Device# show snmp location`	（任意）システムに設定されている場所文字列を表示します。
ステップ 9	**showsnmpchassis** 例： `Device# show snmp chassis`	（任意）システムのシリアル番号を表示します。

SNMP バージョン 1 および 2 の設定

SNMP バージョン 1 および 2 を設定する際には、任意で、コミュニティ ストリングのビューを作成または修正して、SNMP マネージャがアクセスできる MIB オブジェクトを制限することができます。

SNMP バージョン 1 またはバージョン 2 を設定する場合、次の手順を実行します。

前提条件

- SNMP マネージャと SNMP エージェントとの関係を定義する、確立された SNMP コミュニティ ストリング。
- SNMP 通知の受信者として定義されるホスト。

SNMP ビュー レコードの作成または変更

SNMP マネージャがアクセスできる MIB オブジェクトを制限するために、コミュニティ ストリングにビューを割り当てることができます。定義済みビューを使用することも、独自のビューを作成することもできます。定義済みのビューを使用する場合、およびビューを一切使用しない場合は、この手順は飛ばしてください。

SNMP ビュー レコードを作成または変更するには、この作業を実行します。

手順の概要

1. イネーブル化
2. **configureterminal**
3. **snmp-serverview***view-nameoid-tree* {**included** | **excluded**}
4. **nosnmp-serverview***view-nameoid-tree* {**included** | **excluded**}
5. **end**
6. show snmp view

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	イネーブル化 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**snmp-serverview***view-nameoid-tree* {**included** | **excluded**} 例： `Device(config)# snmp-server view mib2 mib-2 included`	ビュー レコードを作成します。 • この例では、MIB-II サブツリー内のすべてのオブジェクトが含まれる mib2 ビューが作成されます。 （注） このコマンドを複数回使用して、同じビュー レコードを作成できます。同じOID値のビュー レコード値が複数回作成される場合、オブジェクト ID の最新のエントリが優先されます。
ステップ 4	**nosnmp-serverview***view-nameoid-tree* {**included** | **excluded**} 例： `Device(config)# no snmp-server view mib2 mib-2 included`	サーバ ビューを削除します。
ステップ 5	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了します。
ステップ 6	show snmp view 例： `Device# show snmp view`	（任意）SNMP に関連付けられている MIB のビューを表示します。

SNMP コミュニティの作成または変更のアクセス コントロール

SNMP マネージャとエージェントと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制御するパスワードのように機能します。ストリングに関連付ける特性を次の中から 1 つ以上指定することもできます。

- エージェントへのアクセスを取得するためにコミュニティ ストリングを使用することを許可された SNMP マネージャの IP アドレスのアクセス リスト

- 特定のコミュニティへのアクセスが可能なすべてのMIBオブジェクトのサブセットを定義する MIB ビュー
- コミュニティへのアクセスが可能なMIBオブジェクトに対する読み書きアクセス権または読み取り専用アクセス権

コミュニティ ストリングを作成または変更するには、この作業を実行します。

手順の概要

1. イネーブル化
2. **configureterminal**
3. **snmp-servercommunity***string* [**view***view-name*] [**ro** | **rw**] [**ipv6***nacl*] [*access-list-number*]
4. **no snmp-server community***string*
5. **end**
6. **showsnmpcommunity**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	イネーブル化 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**snmp-servercommunity***string* [**view***view-name*] [**ro**	**rw**] [**ipv6***nacl*] [*access-list-number*] 例： `Device(config)# snmp-server community comaccess ro 4`	コミュニティ アクセス ストリングを定義します。 • 1 つ以上のコミュニティ ストリングを設定できます。
ステップ 4	**no snmp-server community***string* 例： `Device(config)# no snmp-server community comaccess`	設定からコミュニティ ストリングを削除します。

	コマンドまたはアクション	目的
ステップ 5	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了します。
ステップ 6	**showsnmpcommunity** 例： `Device# show snmp community`	（任意）システムに設定されているコミュニティ アクセス文字列を表示します。

SNMP トラップ操作の受信者の設定

SNMP トラップは、受信者がトラップを受信しても確認応答を送信しないため、信頼性が高くありません。送信者は、トラップが受信されたかどうかを知ることができません。ただし、インフォームを受信した SNMP エンティティは、SNMP 応答 PDU を使用してメッセージに確認応答します。送信側が応答を受信しなかった場合は、再びインフォームを送信できます。このため、インフォームの方が目的の宛先に到達する確実性が高くなります。

トラップと比較すると、インフォームはエージェントおよびネットワークのリソースをより多く消費します。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、メモリ内に保持する必要があります。また、トラップは一度だけ送信されるのに対して、インフォームは数回にわたって送信される場合があります。再試行によってネットワークのトラフィックとオーバーヘッドが増加します。

snmp-serverhost コマンドを入力しなかった場合は、通知が送信されません。SNMP 通知を送信するようにデバイスを設定するには、**snmp-serverhost** コマンドを少なくとも 1 つ入力する必要があります。キーワードを指定せずにこのコマンドを入力すると、ホストに対してすべてのタイプのトラップがイネーブルになります。

複数のホストをイネーブルにするには、各ホストに対して個別に **snmp-serverhost** コマンドを実行する必要があります。コマンドには複数の通知タイプをホストごとに指定できます。

同じホストおよび同じ種類の通知に対して複数の **snmp-serverhost** コマンドを指定した場合は、後に入力されたコマンドによって前のコマンドが上書きされます。最後の **snmp-serverhost** コマンドだけが有効になります。たとえば、ホストに **snmp-serverhostinform** を入力してから、同じホストに別の **snmp-serverhostinform** コマンドを入力した場合は、2 番めのコマンドによって最初のコマンドが置き換えられます。

snmp-serverhost コマンドは、**snmp-serverenable** コマンドと組み合わせて使用します。**snmp-serverenable** コマンドは、どの SNMP 通知がグローバルに送信されるようにするかを指定します。1 つのホストでほとんどの通知を受信する場合は、このホストに対して、少なくとも 1 つの **snmp-serverenable** コマンドと **snmp-serverhost** コマンドをイネーブルにする必要があります。

通知タイプによっては、**snmp-serverenable** コマンドでは制御できません。たとえば、常にイネーブルになっている通知タイプもあれば、別のコマンドでイネーブルにされる通知タイプもあります。たとえば、linkUpDown 通知は **snmptraplink-status** コマンドによって制御されます。このようなタイプの通知には **snmp-serverenable** コマンドは不要です。

notification-type オプションを使用できるかどうかは、デバイスのタイプとそのデバイス上でサポートされている Cisco IOS ソフトウェアの機能によって決まります。たとえば、envmon 通知タイプが使用できるのはシステムに環境モニタが組み込まれている場合のみです。ご使用のシステムで使用できる通知タイプを確認するには、**snmp-serverhost** コマンドの末尾でコマンド ヘルプ（?）を使用します。

SNMP トラップ通知の受信者を設定するには、次の手順を実行します。

手順の概要

1. イネーブル化
2. **configureterminal**
3. **snmp-serverhost***host-id*[**traps** | **informs**] [**version** {**1**| **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port***port-number*] [*notification-type*]
4. **exit**
5. **showsnmphost**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	イネーブル化 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**snmp-serverhost***host-id*[**traps** | **informs**] [**version** {**1**| **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port***port-number*] [*notification-type*] 例： `Device(config)# snmp-server host 172.16.1.27 informs version 2c public alarms`	SNMP 通知をトラップまたは応答要求として送信するかどうか、使用する SNMP のバージョン、通知のセキュリティ レベル（SNMPv3 の場合）、および通知の受信者（ホスト）を指定します。
ステップ 4	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了します。

	コマンドまたはアクション	目的
ステップ 5	**showsnmphost** 例： `Device# show snmp host`	（任意）トラップとして送信される SNMP 通知、および通知のホスト IP アドレスを表示します。

例

次の例は、SNMP 通知に設定されているホスト情報を示します。

```
Device> enable
Device# configure terminal
Device(config)# snmp-server host 10.2.28.1 informs version 2c public
Device(config)# exit
Device# show snmp host

Notification host: 10.2.28.1 udp-port: 162   type: inform
user: public    security model: v2c
traps: 00001000.00000000.00000000
```

SNMP バージョン 3 の設定

SNMPv3 を設定していて、SNMP パケットの処理に SNMPv3 セキュリティ メカニズムを使用する場合は、SNMP グループとパスワード付きのユーザを確立させる必要があります。

次のタスクを実行して、SNMPv3 を設定します。

SNMP サーバ グループ名の指定

SNMPv3 は、セキュリティ モデルです。セキュリティ モデルは、ユーザおよびユーザに属するグループに合わせて設定される認証方式です。

snmp-servergroup コマンドを設定する際、認証やプライバシー アルゴリズムにはデフォルト値はありません。また、デフォルトのパスワードも存在しません。MD5 パスワードを指定する際の詳細については、この **snmp-serveruser** コマンドの説明を参照してください。

新しい SNMP グループまたは SNMP ユーザを SNMP ビューにマッピングするテーブルを指定するには、次の手順を実行します。

手順の概要

1. イネーブル化
2. **configureterminal**
3. **snmp-server group** [*groupname* {**v1** | **v2c** | **v3** [**auth** | **noauth** | **priv**]}] [**read***readview*] [**write***writeview*] [**notify***notifyview*] [**access***access-list*]
4. **exit**
5. **showsnmpgroup**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	イネーブル化 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**snmp-server group** [*groupname* {**v1** | **v2c** | **v3** [**auth** | **noauth** | **priv**]}] [**read***readview*] [**write***writeview*] [**notify***notifyview*] [**access***access-list*] 例： `Device(config)# snmp-server group group1 v3 auth access lmnop`	指定した名前付きアクセス リストのメンバに対して認証をイネーブルにするように SNMP サーバ グループを設定します。 • 次に、*lmnop* という名前のアクセス リストのメンバに対してユーザ認証をイネーブルにするように設定された SNMP サーバ グループ *group1* の例を示します。
ステップ 4	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了します。
ステップ 5	**showsnmpgroup** 例： `Device# show snmp group`	ネットワークの各 SNMP グループに関する情報を表示します。

例

次の例は、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。

```
Device# show snmp group
groupname: ILMI                                 security model:v1
readview : *ilmi                                writeview: *ilmi
notifyview: <no notifyview specified>
row status: active
groupname: ILMI                                 security model:v2c
readview : *ilmi                                writeview: *ilmi
notifyview: <no notifyview specified>
row status: active
groupname: group1                               security model:v3 auth
readview : v1default                            writeview: <no writeview specified>
notifyview: <no notifyview specified>
row status: active                              access-list:lmnop
```

```
groupname: public                               security model:v1
readview : <no readview specified>              writeview: <no writeview specified>
notifyview: <no notifyview specified>
row status: active
```

SNMP サーバ ユーザの設定

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。また、特定のエージェントにリモート ユーザを設定する前に、**snmp-serverengineID** コマンドに remote オプションを使用して SNMP エンジン ID を設定します。リモート エージェントの SNMP エンジン ID は、パスワードから認証とプライバシー ダイジェストを計算する際に必要です。最初にリモート エンジン ID が設定されていない場合、コンフィギュレーション コマンドは失敗します。

privpassword 引数と *auth-password* 引数については、最小の長さが 1 文字で、推奨される長さは 8 文字以上であり、文字と数字の両方を含める必要があります。

SNMP パスワードは、権威 SNMP エンジンの SNMP ID を使用してローカライズされます。インフォームの場合、正規の SNMP エージェントはリモート エンジンです。プロキシ要求またはインフォームを送信できるようにするには、SNMP データベース内のリモート エンジンの SNMP エンジン ID を設定する必要があります。

（注） SNMP ユーザの設定後にエンジン ID を変更した場合、ユーザの削除は許可されません。設定を削除するには、まず、すべての SNMP 設定を再設定する必要があります。

コマンドを設定する際、認証やプライバシー アルゴリズムにはデフォルト値はありません。また、デフォルトのパスワードも存在しません。パスワードの最小の長さは 1 文字ですが、セキュリティのために 8 文字以上にすることを推奨します。パスワードを忘れた場合は回復できないため、ユーザを再設定する必要があります。プレーン テキストのパスワードとローカライズされた MD5 ダイジェストの、どちらも指定できます。

ローカライズされた MD5 または SHA ダイジェストを持っている場合は、プレーン テキストのパスワードではなく、その文字列を指定できます。ダイジェストは aa:bb:cc:dd の形式にする必要があります。aa、bb、および cc は 16 進値です。また、ダイジェストの長さは正確に 16 オクテットにする必要があります。

SNMP グループに新しいユーザを追加するには、次の手順を実行します。

手順の概要

1. イネーブル化
2. **configureterminal**
3. **snmp-server engineID**{**local** *engine-id*| **remote** *ip-address*[**udp-port** *udp-port-number*] [**vrf** *vrf-name*] *engine-id-string*}
4. **snmp-serveruser***usernamegroupname* [**remote***ip-address* [**udp-port***port*]] {**v1** | **v2c** | **v3** [**encrypted**] [**auth** {**md5** | **sha**} *auth-password*]} [**access***access-list*]
5. **exit**
6. **showsnmpuser** [*username*]
7. **show snmp engineID**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	イネーブル化 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**snmp-server engineID**{**local** *engine-id*| **remote** *ip-address*[**udp-port** *udp-port-number*] [**vrf** *vrf-name*] *engine-id-string*} 例： `Device(config)# snmp-server engineID remote 172.12.15.4 udp-port 120 1a2833c0129a`	SNMP エンジン ID を設定します。 • この例では、リモート ユーザの SNMP エンジン ID を設定しています。
ステップ 4	**snmp-serveruser***usernamegroupname* [**remote***ip-address* [**udp-port***port*]] {**v1**	**v2c**	**v3** [**encrypted**] [**auth** {**md5**	**sha**} *auth-password*]} [**access***access-list*] 例： `Device(config)# snmp-server user user1 group1 v3 auth md5 password123`	SNMP グループの新しいユーザを、SNMPv3 グループ「group1」に、ユーザ「user1」、プレーンテキストのパスワード「password123」で設定します。
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 モードを終了し、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	**showsnmpuser** [*username*] 例： `Device# show snmp user user1`	SNMP ユーザの設定した特性について情報を表示します。
ステップ 7	**show snmp engineID** 例： `Device# show snmp engineID`	（任意）SNMP ユーザに対して設定されている SNMP エンジン ID に関する情報を表示します。

例

次に、SNMP user1 に設定された特性についての情報の例を示します。

```
Device# show snmp user user1
User name: user1
Engine ID: 000000090200000000C025808
storage-type: nonvolatile       active access-list: 10
Rowstatus: active
Authentication Protocol: MD5
Privacy protocol: None
Group name: group1
```

デバイスを SNMP マネージャとして設定

SNMP マネージャ プロセスをイネーブルにし、セッション タイムアウト値を設定するには、次の手順を実行します。

手順の概要

1. イネーブル化
2. **configureterminal**
3. **snmp-servermanager**
4. **snmp-servermanagersession-timeout***seconds*
5. **end**
6. **show snmp**
7. **showsnmpsessions[brief]**
8. **showsnmppending**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	イネーブル化 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**snmp-servermanager** 例： `Device(config)# snmp-server manager`	SNMP マネージャをイネーブルにします。
ステップ 4	**snmp-servermanagersession-timeout***seconds* 例： `Device(config)# snmp-server manager session-timeout 30`	（任意）セッションのタイムアウト値を変更します。
ステップ 5	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了します。
ステップ 6	**show snmp** 例： `Device# show snmp`	（任意）SNMP 通信のステータスを表示します。
ステップ 7	**showsnmpsessions[brief]** 例： `Device# show snmp sessions`	（任意）SNMP セッションの状況を表示します。
ステップ 8	**showsnmppending** 例： `Device# show snmp pending`	（任意）保留状態の SNMP 要求の現在のセットを表示します。

例

次に、SNMP 通信のステータスの例を示します。

```
Device# show snmp

Chassis: 01506199
37 SNMP packets input
    0 Bad SNMP version errors
    4 Unknown community name
    0 Illegal operation for community name supplied
    0 Encoding errors
    24 Number of requested variables
    0 Number of altered variables
    0 Get-request PDUs
    28 Get-next PDUs
    0 Set-request PDUs
78 SNMP packets output
    0 Too big errors (Maximum packet size 1500)
    0 No such name errors
    0 Bad values errors
    0 General errors
    24 Response PDUs
    13 Trap PDUs
SNMP logging: enabled
    Logging to 172.17.58.33.162, 0/10, 13 sent, 0 dropped.
SNMP Manager-role output packets
    4 Get-request PDUs
    4 Get-next PDUs
    6 Get-bulk PDUs
    4 Set-request PDUs
    23 Inform-request PDUs
    30 Timeouts
    0 Drops
SNMP Manager-role input packets
    0 Inform response PDUs
    2 Trap PDUs
    7 Response PDUs
    1 Responses with errors
SNMP informs: enabled
    Informs in flight 0/25 (current/max)
    Logging to 172.17.217.141.162
        4 sent, 0 in-flight, 1 retries, 0 failed, 0 dropped
    Logging to 172.17.58.33.162
        0 sent, 0 in-flight, 0 retries, 0 failed, 0 dropped
```

次に、SNMP セッションのステータスの表示例を示します。

```
Device# show snmp sessions

Destination: 172.17.58.33.162, V2C community: public
  Round-trip-times: 0/0/0 (min/max/last)
  packets output
    0 Gets, 0 GetNexts, 0 GetBulks, 0 Sets, 4 Informs
    0 Timeouts, 0 Drops
  packets input
    0 Traps, 0 Informs, 0 Responses (0 errors)
Destination: 172.17.217.141.162, V2C community: public, Expires in 575 secs
  Round-trip-times: 1/1/1 (min/max/last)
  packets output
    0 Gets, 0 GetNexts, 0 GetBulks, 0 Sets, 4 Informs
    0 Timeouts, 0 Drops
 packets input
    0 Traps, 0 Informs, 4 Responses (0 errors)
```

次に、保留状態の SNMP 要求の現在のセットの例を示します。

```
Device# show snmp pending

req id: 47, dest: 172.17.58.33.161, V2C community: public, Expires in 5 secs
req id: 49, dest: 172.17.58.33.161, V2C community: public, Expires in 6 secs
req id: 51, dest: 172.17.58.33.161, V2C community: public, Expires in 6 secs
req id: 53, dest: 172.17.58.33.161, V2C community: public, Expires in 8 secs
```

SNMP マネージャのイネーブル化

SNMPマネージャ プロセスをイネーブルにし、セッション タイムアウト値を設定するには、次の手順を実行します。

手順の概要

1. イネーブル化
2. **configureterminal**
3. **snmp-servermanager**
4. **snmp-servermanagersession-timeout***seconds*
5. **exit**
6. **show snmp**
7. **showsnmpsessions[brief]**
8. **showsnmppending**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	イネーブル化 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**snmp-servermanager** 例： `Device(config)# snmp-server manager`	SNMP マネージャをイネーブルにします。
ステップ 4	**snmp-servermanagersession-timeout***seconds* 例： `Device(config)# snmp-server manager session-timeout 30`	（任意）セッションのタイムアウト値を変更します。
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 モードを終了します。

	コマンドまたはアクション	目的
ステップ 6	**show snmp** 例： `Device# show snmp`	（任意）SNMP 通信のステータスを表示します。
ステップ 7	**showsnmpsessions[brief]** 例： `Device# show snmp sessions`	（任意）SNMPセッションのステータスを表示します。
ステップ 8	**showsnmppending** 例： `Device# show snmp pending`	（任意）保留状態の SNMP 要求の現在のセットを表示します。

例

次に、SNMP 通信のステータスの例を示します。

```
Device# show snmp
Chassis: 01506199
37 SNMP packets input
    0 Bad SNMP version errors
    4 Unknown community name
    0 Illegal operation for community name supplied
    0 Encoding errors
    24 Number of requested variables
    0 Number of altered variables
    0 Get-request PDUs
    28 Get-next PDUs
    0 Set-request PDUs
78 SNMP packets output
    0 Too big errors (Maximum packet size 1500)
    0 No such name errors
    0 Bad values errors
    0 General errors
    24 Response PDUs
    13 Trap PDUs
SNMP logging: enabled
    Logging to 172.17.58.33.162, 0/10, 13 sent, 0 dropped.
SNMP Manager-role output packets
    4 Get-request PDUs
    4 Get-next PDUs
    6 Get-bulk PDUs
    4 Set-request PDUs
    23 Inform-request PDUs
    30 Timeouts
    0 Drops
SNMP Manager-role input packets
    0 Inform response PDUs
    2 Trap PDUs
    7 Response PDUs
    1 Responses with errors
SNMP informs: enabled
    Informs in flight 0/25 (current/max)
    Logging to 172.17.217.141.162
```

```
        4 sent, 0 in-flight, 1 retries, 0 failed, 0 dropped
    Logging to 172.17.58.33.162
        0 sent, 0 in-flight, 0 retries, 0 failed, 0 dropped
```

次に、SNMP セッションのステータスの表示例を示します。

```
Device# show snmp sessions
Destination: 172.17.58.33.162, V2C community: public
  Round-trip-times: 0/0/0 (min/max/last)
  packets output
    0 Gets, 0 GetNexts, 0 GetBulks, 0 Sets, 4 Informs
    0 Timeouts, 0 Drops
  packets input
    0 Traps, 0 Informs, 0 Responses (0 errors)
Destination: 172.17.217.141.162, V2C community: public, Expires in 575 secs
  Round-trip-times: 1/1/1 (min/max/last)
  packets output
    0 Gets, 0 GetNexts, 0 GetBulks, 0 Sets, 4 Informs
    0 Timeouts, 0 Drops
 packets input
    0 Traps, 0 Informs, 4 Responses (0 errors)
```

次に、保留状態の SNMP 要求の現在のセットの例を示します。

```
Device# show snmp pending
req id: 47, dest: 172.17.58.33.161, V2C community: public, Expires in 5 secs
req id: 49, dest: 172.17.58.33.161, V2C community: public, Expires in 6 secs
req id: 51, dest: 172.17.58.33.161, V2C community: public, Expires in 6 secs
req id: 53, dest: 172.17.58.33.161, V2C community: public, Expires in 8 secs
```

SNMP エージェント シャットダウン メカニズムのイネーブル化

SNMP パケットを使用するネットワーク管理ツールでは、仮想端末およびコンソール上のユーザに対してメッセージを送信できます。このファシリティは **send** EXEC コマンドと似た方法で動作します。ただし、ユーザに対してメッセージを発行するための SNMP 要求では、メッセージの送信後に実行するアクションも指定します。このようなアクションとしては、シャットダウン要求があります。システムは、シャットダウンされた後、通常はリロードされます。ネットワークからリロードを発生させることができるというのは強力な機能であるため、これは **snmp-serversystem-shutdown** グローバル コンフィギュレーション コマンドで保護されています。このコマンドを発行しない場合、シャットダウン メカニズムがイネーブルになりません。

SNMP エージェントのシャットダウン メカニズムをイネーブルにするには、次の手順を実行します。

手順の概要

1. イネーブル化
2. **configureterminal**
3. **snmp-serversystem-shutdown**
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	イネーブル化 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**snmp-serversystem-shutdown** 例： `Device(config)# snmp-server system-shutdown`	SNMP メッセージのリロード機能を使用して、システムのシャットダウンをイネーブルにします。
ステップ 4	end 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了します。

SNMP エージェント パケットの最大サイズの定義

SNMP エージェントが要求の受信または応答の生成を行うときに許容される最大パケット サイズを定義できます。

許容される最大パケット サイズを設定するには、次の手順を実行します。

手順の概要

1. イネーブル化
2. **configureterminal**
3. **snmp-serverpacketsize***byte-count*
4. **exit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	イネーブル化 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**snmp-serverpacketsize***byte-count* 例： `Device(config)# snmp-server packetsize 512`	最大パケット サイズを確立します。
ステップ 4	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

SNMP を介して使用される TFTP サーバ数の制限

アクセス リストを使用して、SNMP を介してコンフィギュレーション ファイルの保存およびロードに使用される TFTP サーバの数を制限できます。TFTP サーバの使用をこのような方法で制限すると、システム リソースを保護し、管理のための操作を集中化することができます。

TFTP サーバの数を制限するには、次の手順を実行します。

手順の概要

1. イネーブル化
2. **configureterminal**
3. **snmp-servertftp-server-list***number*
4. **exit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	イネーブル化 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**snmp-servertftp-server-list***number* 例： `Device(config)# snmp-server tftp-server-list 12`	アクセスリストで、SNMP経由でのコンフィギュレーション ファイルのサーバへのコピーに使用される TFTP サーバの数を制限します。
ステップ 4	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

トラブルシューティングのヒント

トラブルシューティングのためにリアルタイムでの SNMP トラップ アクティビティをモニタするには、SNMP **debug** コマンド（ **debug snmp packet** EXEC コマンドなど）を使用します。SNMP **debug** コマンドに関する詳細については、『*Cisco IOS Debug Command Reference*』を参照してください。

SNMP エージェントのディセーブル化

SNMP エージェントのバージョンをディセーブルにするには、次の手順を実行します。

手順の概要

1. イネーブル化
2. **configureterminal**
3. **nosnmp-server**
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	イネーブル化 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**nosnmp-server** 例： `Device(config)# no snmp-server`	SNMPエージェントの運用をディセーブルにします。
ステップ 4	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了します。

SNMP 通知の設定

SNMP トラップまたはインフォームを送信するようにデバイスを設定するには、ここで説明する手順を実行します。

（注） 多くの snmp-server コマンドでは、そのコマンド構文で **traps** というキーワードが使用されます。トラップまたはインフォームを指定するオプションがコマンドに指定されていない限り、キーワード **traps** を使用してトラップ、インフォーム、または両方を指定する必要があります。**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のどちらとして SNMP 通信を送信するか指定します。インフォームを使用するには、SNMP マネージャ（SNMP プロキシ マネージャとも呼ばれる）が使用可能で、デバイス上でイネーブルになっている必要があります。以前は、SNMP マネージャを使用できるのは Cisco IOS PLUS イメージだけでした。ただし、SNMP マネージャは SNMP をサポートするすべての Cisco ソフトウェア リリースで現在使用可能です。各 Cisco ソフトウェア リリースでの SNMP マネージャのサポートに関する情報を参照するには、Cisco Feature Navigator を使用してください。Cisco Feature Navigator には、http://www.cisco.com/go/cfn [英語] からアクセスします。

SNMP 通知を送信するようにデバイスを設定する

トラップまたはインフォームをホストに送信するようにデバイスを設定するには、この手順を実行します。

手順の概要

1. イネーブル化
2. **configureterminal**
3. **snmp-server engineID remote** *remote-ip-address remote-engineID*
4. **snmp-serveruser***usernamegroupname* [**remote***host* [**udp-port***port*] {**v1** | **v2c** | **v3** [**encrypted**] [**auth** {**md5** | **sha**} *auth-password*]} [**access***access-list*]
5. **snmp-servergroup***groupname* {**v1** | **v2c** | **v3**{**auth**| **noauth** | **priv**}} [**read***readview*] [**write***writeview*] [**notify***notifyview*] [**access***access-list*]
6. **snmp-serverhost***host*[**traps** | **informs**] [**version**{**1**|**2c**|**3**[**auth**|**noauth**|**priv**]}] *community-string*[*notification-type*]
7. **snmp-serverenabletraps**[*notification-type* [*notification-options*]]
8. 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	イネーブル化 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**snmp-server engineID remote** *remote-ip-address remote-engineID* 例： `Device(config)# snmp-server engineID remote 172.16.20.3 80000009030000B064EFE100`	SNMP エンジン ID を指定し、172.16.20.3 にあるリモート デバイスとの SNMP 通信のための VRF 名 traps-vrf を設定します。
ステップ 4	**snmp-serveruser***username groupname* [**remote***host* [**udp-port***port*] {**v1** | **v2c** | **v3** [**encrypted**] [**auth** {**md5** | **sha**} *auth-password*]} [**access***access-list*] 例： `Device(config)# snmp-server user abcd public v3 encrypted auth md5 cisco123`	ローカル ユーザまたはリモート ユーザを SNMP グループに設定します。 （注） アドレスのリモート ユーザを設定するには、まずそのリモート ホストのエンジン ID を設定する必要があります。この制限は、これらのコマンドの設計として課されたものです。ホストよりも前にユーザを設定しようとすると、警告メッセージが表示され、コマンドは実行されません。**snmp-serverengineidremote** コマンドを使用して、リモート ホストのエンジン ID を指定します。
ステップ 5	**snmp-servergroup***groupname* {**v1** | **v2c** | **v3**{**auth**| **noauth** | **priv**}} [**read***readview*] [**write***writeview*] [**notify***notifyview*] [**access***access-list*] 例： `Device(config)# snmp-server group GROUP1 v2c auth read viewA write viewA notify viewB`	SNMP グループを設定します。
ステップ 6	**snmp-serverhost***host*[**traps** | **informs**] [**version**{**1**|**2c**|**3**[**auth**|**noauth**|**priv**]}] *community-string*[*notification-type*] 例： `Device(config)# snmp-server host example.com informs version 3 public`	SNMP 通知をトラップまたは応答要求として送信するかどうか、使用する SNMP のバージョン、通知のセキュリティ レベル（SNMPv3 の場合）、および通知の受信者（ホスト）を指定します。 • **snmp-serverhost** コマンドでは、どのホストで SNMP 通知を受信するか、および通知がトラップとインフォームのいずれで送信されるようにするかを指定します。

	コマンドまたはアクション	目的
ステップ 7	**snmp-serverenabletraps**[*notification-type* [*notification-options*]] 例： `Device(config)# snmp-server enable traps bgp`	トラップまたはインフォームの送信をイネーブルにして、送信される通知のタイプを指定します。 • *notification-type* が指定されていない場合は、サポートされているすべての通知がデバイスでイネーブルになります。 • デバイスで使用可能な通知を確認するには、**snmp-serverenabletraps?** コマンドを入力します。 • **snmp-serverenabletraps** コマンドは、指定された通知タイプ（Border Gateway Protocol（BGP）トラップ、設定トラップ、エンティティ トラップ、ホットスタンバイ デバイス プロトコル（HSDP）トラップなど）の生成メカニズムをグローバルにイネーブルにします。
ステップ 8	end 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

通知操作値の変更

ソース インターフェイス、各ホストのメッセージ（パケット）キューの長さ、再送信の間隔などにデフォルト値以外の値を指定できます。

必要に応じて通知オペレーション値を変更するには、次の手順を実行します。

手順の概要

1. イネーブル化
2. **configureterminal**
3. **snmp-servertrap-source***interface*
4. **snmp-serverqueue-length***length*
5. **snmp-servertrap-timeout***seconds*
6. **snmp-serverinforms** [**retries** *retries*] [**timeout** *seconds*] [**pending** *pending*]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	イネーブル化	特権 EXEC モードをイネーブルにします。

	コマンドまたはアクション	目的
	例： `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	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**snmp-servertrap-source***interface* 例： `Device(config)# snmp-server trap-source FastEthernet 2/1`	スロット 2、ポート 1 にあるファストイーサネット インターフェイスの IP アドレスをすべての SNMP 通知の送信元として設定します。
ステップ 4	**snmp-serverqueue-length***length* 例： `Device(config)# snmp-server queue-length 50`	各通知のメッセージ キューの長さを設定します。 • この例は、キューの長さを 50 エントリに設定しています。
ステップ 5	**snmp-servertrap-timeout***seconds* 例： `Device(config)# snmp-server trap-timeout 30`	再送信キューにある通知を再送信する頻度を定義します。
ステップ 6	**snmp-serverinforms** [**retries** *retries*] [**timeout** *seconds*] [**pending** *pending*] 例： `Device(config)# snmp-server informs retries 10 timeout 30 pending 100`	インフォーム固有のオペレーション値を設定します。 • この例は、インフォームを再送信する最大回数、再送信するまでに確認応答を待って待機する秒数、および一度に確認応答を待てるインフォームの最大数を設定しています。

個々の RFC 1157 SNMP トラップの制御

authenticationFailure、linkUp、linkDown、warmStart、および coldStart 通知の各タイプをイネーブルにするには、次の手順を実行します。

手順の概要

1. イネーブル化
2. **configureterminal**
3. **snmp-serverenabletrapssnmp** [**authentication**] [**linkup**] [**linkdown**] [**warmstart**] [**coldstart**]
4. **interface** *type slot/port*
5. **nosnmp-serverlink-status**
6. **end**
7. **end**
8. **show snmp mib ifmibtraps**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	イネーブル化 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**snmp-serverenabletrapssnmp** [**authentication**] [**linkup**] [**linkdown**] [**warmstart**] [**coldstart**] 例： `Device(config)# snmp-server enable traps snmp`	RFC 1157 汎用トラップをイネーブルにします。 • オプションのキーワードを一切指定せずに使用した場合、authenticationFailure、linkUp、linkDown、warmStart、および coldStart の各トラップをイネーブルにします。 • キーワード指定で使用した場合は、指定したタイプのトラップのみがイネーブルになります。たとえば、すべてのインターフェイスに対して linkUp と linkDown の SNMP トラップまたは SNMP インフォームだけをグローバルにイネーブルにするには、このコマンドの **snmp-serverenabletrapssnmplinkuplinkdown** という形式を使用します。
ステップ 4	**interface** *type slot/port* 例： `Device(config)# interface FastEthernet 0/0`	該当インターフェイスのインターフェイス 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
		（注） ダイヤラなどの個別のインターフェイスに対してSNMPトラップをイネーブルにするには、インターフェイスコンフィギュレーション モードで **snmptraplink-statuspermitduplicates** コマンドを使用します。たとえば、ダイヤラインターフェイスコンフィギュレーション モードを開始して、インターフェイスタイプとしてダイヤラを開始します。
ステップ 5	**nosnmp-serverlink-status** 例： `Device(config-if)# no snmp-server link-status`	すべての標準インターフェイスの linkUp 通知と linkDown 通知の送信をディセーブルにします。
ステップ 6	**end** 例： `Device(config-if)# end`	インターフェイスコンフィギュレーションモードを終了します。
ステップ 7	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了し、特権EXECモードに戻ります。
ステップ 8	**show snmp mib ifmibtraps** 例： `Device# show snmp mib ifmib traps`	

例

次の例は、システムに設定されているすべてのインターフェイスの linkup トラップと linkdown トラップのステータスを表示します。

```
Device# show snmp mib ifmib traps

ifDescr   ifindex  TrapStatus
---------------------------------
FastEthernet  3/6 14  enabled
FastEthernet  3/19 27  enabled
GigabitEthernet  5/1 57  enabled
unrouted VLAN  1005 73  disabled
FastEthernet  3/4 12  enabled
FastEthernet  3/39 47  enabled
FastEthernet  3/28 36  enabled
FastEthernet  3/48 56  enabled
unrouted VLAN  1003 74  disabled
```

```
FastEthernet  3/2 10  enabled
Tunnel   0 66  enabled
SPAN RP Interface  64  disabled
Tunnel   10 67  enabled
FastEthernet  3/44 52  enabled
GigabitEthernet  1/3 3  enabled
FastEthernet  3/11 19  enabled
FastEthernet  3/46 54  enabled
GigabitEthernet  1/1 1  enabled
FastEthernet  3/13 21  enabled
unrouted VLAN  1 70  disabled
GigabitEthernet  1/4 4  enabled
FastEthernet  3/9 17  enabled
FastEthernet  3/16 24  enabled
FastEthernet  3/43 51  enabled
```

SNMP 通知ログ オプションの設定

SNMP 通知ログ オプションを設定するには、次の手順を実行します。これらのオプションで、ログ サイズとタイミングの値を制御できます。変更しないまま置いておくと、SNMP ログが非常に大きく、長くなる場合があります。

手順の概要

1. イネーブル化
2. **configureterminal**
3. **snmpmibnotification-logdefault**
4. **snmpmibnotification-logglobalageout***seconds*
5. **snmpmibnotification-logglobalsize***size*
6. **end**
7. **showsnmpmibnotification-log**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	イネーブル化 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**snmpmibnotification-logdefault** 例： `Device(config)# snmp mib notification-log default`	名前の付いていない SNMP 通知ログを作成します。

	コマンドまたはアクション	目的
ステップ 4	**snmpmibnotification-logglobalageout***seconds* 例： `Device(config)# snmp mib notification-log globalageout 20`	SNMP 通知ログ エントリがシステム メモリ内に保持され続ける最大時間を設定します。 • この例では、SNMP 通知ログ内の 20 分以上前に記録されたエントリを削除するようにシステムを設定しています。
ステップ 5	**snmpmibnotification-logglobalsize***size* 例： `Device(config)# snmp mib notification-log globalsize 600`	すべてSNMP通知ログに保存できるエントリの最大数を設定します。
ステップ 6	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了します。
ステップ 7	**showsnmpmibnotification-log** 例： `Device# show snmp mib notification-log`	ローカルSNMP通知ロギングの状態に関する情報を表示します。

例

次に、ローカル SNMP 通知ロギングの状態に関する情報の例を示します。

```
Device# show snmp mib notification-log

GlobalAgeout 20, GlobalEntryLimit 600
Total Notifications logged in all logs 0
Log Name"", Log entry Limit 600, Notifications logged 0
Logging status enabled
Created by cli
```

インターフェイス インデックス表示の設定とインターフェイス インデックスおよびロング ネーム サポート

インターフェイス インデックスの表示では、SNMP の上級ユーザが管理対象エージェントで直接インターフェイス登録に関する情報を表示できます。外部 NMS は必要ありません。

インターフェイスのロング エイリアス名の設定では、ユーザが、ifAlias（MIB 内で定義されるオブジェクトで、長さは 64 までに制限されています）を最大 255 バイトに設定できるようにできます。

はじめる前に

システム上で SNMP がイネーブルになっている必要があります。

インターフェイスインデックス表示とインターフェイスエイリアスロング名サポートの機能は、すべての Cisco プラットフォームでサポートされているわけではありません。プラットフォームのサポートおよびソフトウェア 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。

64 文字より長い ifAlias 値を保持するように IF-MIB を設定し、インターフェイスの ifAlias 値を設定するには、次の手順を実行します。

（注） ifAlias の説明が 64 文字よりも長いかどうかを確認するには、NMS から ifMIB ifAlias 変数の SNMP MIB ウォークを実行し、説明全体が ifXEntry.18 の値内に表示されることを確認します。インターフェイスの説明は、**moresystem:runningconfig** 特権 EXEC モード コマンドからの出力にも表示されます。

手順の概要

1. イネーブル化
2. **configureterminal**
3. **snmpifmibifaliaslong**
4. **interface***typenumber*
5. **description***text-string*
6. **end**
7. **showsnmpmib**
8. **showsnmpmibifmibifindex** [*typenumber*] [**detail**] [**free-list**]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	イネーブル化 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	**snmpifmibifaliaslong** 例： `Device(config)# snmp ifmib ifalias long`	ネットワーク管理システムに 64 文字より長い ifAlias 値を返すように、システム上のインターフェイス MIB（IF-MIB）を設定します。 • **snmpifmibifaliaslong** コマンドを使用して ifAlias 値を設定していないと、ifAlias の説明は 64 文字までに制限されます。
ステップ 4	**interface***typenumber* 例： `Device(config)# interface ethernet 2/4`	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。 • このコマンドの形式は、設定しているインターフェイスによって変わります。
ステップ 5	**description***text-string* 例： `Device(config)# description This text string description can be up to 256 characters long`	指定されたインターフェイスの自由形式の説明を設定します。 • この説明の長さは最大 240 文字で、IF-MIB 内の ifAlias オブジェクト値として保存できます。 • **snmpifmibifaliaslong** コマンドを使用して ifAlias 値を設定していなかった場合、インターフェイスの説明 **description** コマンドを使用して 64 文字より多い説明に設定されている場合でも、SNMP の set 操作および get 操作の ifAlias の説明は 64 文字までに制限されます。
ステップ 6	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了します。
ステップ 7	**showsnmpmib** 例： `Device# show snmp mib`	システムに登録されている MIB モジュール インスタンス識別子のリストを表示します。 • 表示される内容が非常に長くなる場合があります。
ステップ 8	**showsnmpmibifmibifindex** [*typenumber*] [**detail**] [**free-list**] 例： `Device# show snmp mib ifmib ifindex Ethernet 2/0`	システム上で登録された、すべてのインターフェイスまたは指定されたインターフェイスのインターフェイス MIB ifIndex 値を表示します。

例

次に、システム上で登録された MIB モジュール インスタンス ID の例を示します。表示される内容が非常に長くなる場合があります。ここでは、ごく一部だけを示します。

```
Device# show snmp mib
system.1
system.2
sysUpTime
system.4
system.5
system.6
system.7
system.8
sysOREntry.2
sysOREntry.3
sysOREntry.4
interfaces.1
ifEntry.1
ifEntry.2
ifEntry.3
ifEntry.4
ifEntry.5
ifEntry.6
ifEntry.7
ifEntry.8
ifEntry.9
ifEntry.10
ifEntry.11
 --More--
captureBufferEntry.2
captureBufferEntry.3
captureBufferEntry.4
captureBufferEntry.5
captureBufferEntry.6
captureBufferEntry.7
capture.3.1.1
eventEntry.1
eventEntry.2
eventEntry.3
eventEntry.4
eventEntry.5
eventEntry.6
eventEntry.7
logEntry.1
logEntry.2
logEntry.3
logEntry.4
rmon.10.1.1.2
rmon.10.1.1.3
rmon.10.1.1.4
rmon.10.1.1.5
rmon.10.1.1.6
rmon.10.1.1.7
rmon.10.2.1.2
rmon.10.2.1.3
rmon.10.3.1.2
```

次の例は、指定されたインターフェイスについて、システムに登録されているインターフェイス MIB ifIndex 値を表示します。

```
Device# show snmp mib ifmib ifindex Ethernet 2/0
Ethernet2/0: Ifindex = 2
```

次の例は、すべてのインターフェイスについて、システムに登録されているインターフェイス MIB ifIndex 値を表示します。

```
Device# show snmp mib ifmib ifindex
ATM1/0: Ifindex = 1
ATM1/0-aal5 layer: Ifindex = 12
ATM1/0-atm layer: Ifindex = 10
ATM1/0.0-aal5 layer: Ifindex = 13
```

```
ATM1/0.0-atm subif: Ifindex = 11
ATM1/0.9-aal5 layer: Ifindex = 32
ATM1/0.9-atm subif: Ifindex = 31
ATM1/0.99-aal5 layer: Ifindex = 36
ATM1/0.99-atm subif: Ifindex = 35
Ethernet2/0: Ifindex = 2
Ethernet2/1: Ifindex = 3
Ethernet2/2: Ifindex = 4
Ethernet2/3: Ifindex = 5
Null0: Ifindex = 14
Serial3/0: Ifindex = 6
Serial3/1: Ifindex = 7
Serial3/2: Ifindex = 8
Serial3/3: Ifindex = 9
```

インターフェイス インデックス パーシステンスの設定

ここでは、インターフェイスインデックスパーシステンスを設定するための次の手順を説明します。

ifIndex パーシステンスのグローバルなイネーブル化とディセーブル化

ifIndex パーシステンスをグローバルにイネーブルにするには、次の手順を実行します。

はじめる前に

ここで説明する設定手順では、使用しているルーティング デバイス上ですでに SNMP が設定してあり、SNMP を使用して、Cisco コマンドライン インターフェイスや NMS アプリケーションによるネットワーク アクティビティ監視を行っていることが前提となっています。

インターフェイス専用の ifIndex パーシステンス コマンド（**snmpifindexpersistence**）は、サブインターフェイスには使用できません。インターフェイスに適用されるコマンドは、そのインターフェイスに関連するすべてのサブインターフェイスに自動的に適用されます。

テストによると、この機能によって、インターフェイスごとに約 25 バイトの NVRAM ストレージが使用されます。低速の CPU では、プラットフォーム上でブート遅延が発生する可能性があります。

（注） ifIndex 値の一貫性を確保するために、ifIndex パーシステンス コマンドを入力した後は **copyrunning-configstartup-config** EXEC モード コマンドを使用して設定を保存する必要があります。

手順の概要

1. イネーブル化
2. **configureterminal**
3. **snmp-serverifindexpersist**
4. **nosnmp-serverifindexpersist**
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	イネーブル化 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**snmp-serverifindexpersist** 例： `Device(config)# snmp-server ifindex persist`	リブート後も保持される ifIndex 値をグローバルにイネーブルにします。
ステップ 4	**nosnmp-serverifindexpersist** 例： `Device(config)# no snmp-server ifindex persist`	グローバル ifIndex パーシステンスをディセーブルにします。
ステップ 5	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了します。

特定のインスタンスでの ifIndex パーシステンスのイネーブル化およびディセーブル化

ifIndex パーシステンスを指定したインターフェイス上でだけ設定するには、次の手順を実行します。

ヒント　インターフェイスで ifIndex パーシステンスのためにグローバル コンフィギュレーション設定を使用する場合、特定のインターフェイスで **snmpifindexclear** コマンドを使用します。このコマンドでは、特定のインターフェイスに対して以前に開始された ifIndex コンフィギュレーション コマンドをクリアします。

手順の概要

1. イネーブル化
2. **configureterminal**
3. **interface***typeslot/port*
4. **snmpifindexpersist**
5. **nosnmpifindexpersist**
6. **end**
7. **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	イネーブル化 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface***typeslot/port* 例： `Device(config)# interface FastEthernet 0/1`	指定したインターフェイスに対してインターフェイス コンフィギュレーション モードを開始します。 （注） インターフェイス コマンドの構文は使用しているプラットフォームにより異なることに注意してください。
ステップ 4	**snmpifindexpersist** 例： `Device(config-if)# snmp ifindex persist`	指定したインターフェイス上での ifIndex 値のリブートにまたがった一貫性をイネーブルにします。
ステップ 5	**nosnmpifindexpersist** 例： `Device(config-if)# no snmp ifindex persist`	指定したインターフェイス上での ifIndex 値のリブートにまたがった一貫性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 6	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーション モードを終了します。
ステップ 7	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了します。

VPN のための SNMP サポートの設定

ここでは、VPN のために SNMP を設定する方法を説明します。VPN のための SNMP サポートの機能は、ユーザが SNMP エージェントとマネージャを特定の VRF と関連付けることを可能にするコンフィギュレーション コマンドを提供します。指定された VRF を使用して、SNMP トラップと SNMP インフォーム、およびエージェントとマネージャ間の応答の送信が実行されます。VRF が指定されなかった場合は、その VPN のデフォルトのルーティング テーブルが使用されます。

VPN のサポートにより、ユーザが特定の VPN セットからの SNMP 要求のみを受け取るように SNMP エージェントを設定できます。この設定では、プロバイダーが顧客に、すべてのユーザの VPN デバイスを管理できるネットワーク管理サービスを提供できます。

（注）
- この機能はすべての Cisco プラットフォームでサポートされるわけではありません。Cisco Feature Navigator を使用すると、プラットフォーム、および Cisco IOS ソフトウェア イメージの各サポート情報を検索できます。
- すべての MIB が VPN に対応しているわけではありません。VPN 対応の MIB のリストを見るには、**showsnmpmibcontext** コマンドを使用してください。VPN 対応 MIB の詳細については、設定に関する章「*SNMP Support over VPNs*：コンテキストベース アクセス コントロール」を参照してください。

特定の VPN のための SNMP サポートを設定するには、このタスクを実行します。

手順の概要

1. イネーブル化
2. **configureterminal**
3. **snmp-serverhost***host-address*[**vrf***vrf-name*] [**traps** | **informs**] [**version** {**1**| **2c**| **3** [**auth** | **noauth** |**priv**]}] *community-string* [**udp-port***port*] [*notification-type*]
4. **snmp-serverengineIDremote***ip-address* [**udp-port***udp-port-number*] [**vrf***vrf-name*] *engineid-string*
5. **exit**
6. **showsnmphost**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	イネーブル化 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**snmp-serverhost***host-address*[**vrf***vrf-name*] [**traps**	**informs**] [**version** {**1**| **2c**| **3** [**auth**	**noauth** |**priv**]}] *community-string* [**udp-port***port*] [*notification-type*] 例： `Device(config)# snmp-server host example.com public vrf trap-vrf`	SNMP 通知動作の受信先を指定し、SNMP 通知の送信に使用される VRF テーブルを指定します。

ステップ 4	**snmp-serverengineIDremote***ip-address* [**udp-port***udp-port-number*] [**vrf***vrf-name*] *engineid-string*
	
	例：
	`Device(config)# snmp-server engineID remote 172.16.20.3 vrf traps-vrf`
	
	例：
	`80000009030000B064EFE100`
		リモート SNMP ユーザのために特定の VPN 上で SNMP を設定する場合、デバイス上のリモート SNMP エンジンの名前を設定します。

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 モードを終了します。

	コマンドまたはアクション	目的
ステップ 6	**showsnmphost** 例： `Device# show snmp host`	（任意）SNMP 設定を表示し、VPN 機能のための SNMP サポートが適切に設定されていることを確認します。

SNMP を使用するイベント MIB の設定

イベント MIB は、SNMP を使用して直接設定できます。この手順では、1 分おきにすべてのインターフェイスの ifInOctets のデルタ値をモニタするようにイベント MIB を設定します。指定したしきい値を超えるサンプルがあった場合、トラップ通知が送信されます。

イベント MIB に関連付けられている Cisco ソフトウェア設定手順はありません。イベント MIB 機能の設定はすべて、SNMP を使用して、アプリケーションを通じて実行する必要があります。ここでは、外部デバイス上のネットワーク管理アプリケーションを使用したサンプル設定セッションを示します。お使いの Cisco ルーティング デバイスでの SNMP の設定については、「その他の参考資料」セクションを参照してください。

イベント MIB 機能の設定はすべて、SNMP を使用して、アプリケーションを通じて実行する必要があります。以降では、Sun のワークステーションで使用できる SNMP 調査ツールを使用したイベント MIB の設定方法をステップ バイ ステップで示します。ここで示す **setany** コマンドは、SNMP アプリケーションを使用して実行されます。

（注） Cisco コマンドライン インターフェイス コマンドではありません。使用しているルーティング デバイス上で SNMP が設定されていることを前提としています。

この設定での目的は、すべてのインターフェイスの ifInOctets をモニタすることです。1 分おきにすべてのインターフェイスの ifInOctets のデルタ値をモニタするようにイベント MIB を設定します。指定したしきい値 30 を超えるサンプルがあった場合、トラップ通知が送信されます。

次の例には 5 つの部分があります。

トリガー テーブルでのトリガーの設定

トリガー テーブル内のトリガーを設定するには、次の手順を実行します。

手順の概要

1. **setany-v2c$ADDRESSprivatemteTriggerEntryStatus.4.106.111.104.110.1-i5**
2. **setany-v2c$ADDRESSprivatemteTriggerValueID.4.106.111.104.110.1-d1.3.6.1.2.1.2.2.1.10**
3. **setany-v2c$ADDRESSprivatemteTriggerValueIDWildcard.4.106.111.104.110.1-i1**
4. **setany-v2c$ADDRESSprivatemteTriggerTest.4.106.111.104.110.1-o'20'**
5. **setany-v2c$ADDRESSprivatemteTriggerFrequency.4.106.111.104.110.1-g60**
6. **setany-v2c$ADDRESSprivatemteTriggerSampleType.4.106.111.104.110.1-i2**
7. **setany-v2c$ADDRESSprivatemteTriggerEnabled.4.106.111.104.110.1-i1**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**setany-v2c$ADDRESSprivatemteTriggerEntryStatus.4.106.111.104.110.1-i5**	トリガー テーブルに、mteOwner が john、トリガー名が 1 のトリガー行を作成します。 • インデックスは、10 進数表記の ASCII 値、john.1 で与えられます。
ステップ 2	**setany-v2c$ADDRESSprivatemteTriggerValueID.4.106.111.104.110.1-d1.3.6.1.2.1.2.2.1.10**	mteTriggerValueID を監視される OID に設定します。 • この例では、モニタされる OID は、ifInOctets です。
ステップ 3	**setany-v2c$ADDRESSprivatemteTriggerValueIDWildcard.4.106.111.104.110.1-i1**	オブジェクトがワイルドカードで参照されることを示すために、mteTriggerValueIDWildcard を TRUE に設定します。
ステップ 4	**setany-v2c$ADDRESSprivatemteTriggerTest.4.106.111.104.110.1-o'20'**	mteTriggerTest をしきい値に設定します。
ステップ 5	**setany-v2c$ADDRESSprivatemteTriggerFrequency.4.106.111.104.110.1-g60**	mteTriggerFrequency を 60 に設定します。これで、60 秒おきに ifInOctets がモニタされます。

	コマンドまたはアクション	目的
ステップ 6	**setany-v2c$ADDRESSprivatemteTriggerSampleType.4.106.111.104.110.1-i2**	サンプル タイプをデルタに設定します。
ステップ 7	**setany-v2c$ADDRESSprivatemteTriggerEnabled.4.106.111.104.110.1-i1**	トリガーをイネーブルにします。

イベント テーブルでのイベントの作成

イベント テーブルにイベントを作成するには、次の手順を実行します。

手順の概要

1. **setany-v2c$ADDRESSprivatemteEventEntryStatus.4.106.111.104.110.101.118.101.110.116-i5**
2. **setany-v2c$ADDRESSprivatemteEventEnabled.4.106.111.104.110.101.118.101.110.116-i1**
3. **setany-v2c$ADDRESSprivatemteEventEntryStatus.4.106.111.104.110.101.118.101.110.116-i1**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**setany-v2c$ADDRESSprivatemteEventEntryStatus.4.106.111.104.110.101.118.101.110.116-i5**	イベントテーブルに行を作成します。 • ここでも、mteOwner は john で、イベントは mteEventName です。 • デフォルト アクションでは、通知が送信されます。
ステップ 2	**setany-v2c$ADDRESSprivatemteEventEnabled.4.106.111.104.110.101.118.101.110.116-i1**	イベント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	**setany-v2c$ADDRESSprivatemteEventEntryStatus.4.106.111.104.110.101.118.101.110.116-i1**	EventRow をアクティブにします。

トリガー テーブルでのトリガーのしきい値の設定とアクティブ化

トリガー テーブル内のトリガーしきい値を設定するには、次の手順を実行します。

手順の概要

1. **setany-v2c$ADDRESSprivatemteTriggerThresholdRising.4.106.111.104.110.1-i30**
2. **setany-v2c$ADDRESSprivatemteTriggerThresholdRisingEventOwner.4.106.111.104.110.1-D"owner"**
3. **setany-v2c$ADDRESSprivatemteTriggerEntryStatus.4.106.111.104.110.1-i1**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**setany-v2c$ADDRESSprivatemteTriggerThresholdRising.4.106.111.104.110.1-i30**	上昇しきい値を 30 に設定します。トリガーしきい値のテーブル内には、すでに john.1 用の行が存在します。
ステップ 2	**setany-v2c$ADDRESSprivatemteTriggerThresholdRisingEventOwner.4.106.111.104.110.1-D"owner"** 例： `setany -v2c $ADDRESS private mteTriggerThresholdRisingEvent.4.106.111.104.110.1 -D "event"`	イベントテーブル内の、実行されるアクションを指定するエントリを示します。
ステップ 3	**setany-v2c$ADDRESSprivatemteTriggerEntryStatus.4.106.111.104.110.1-i1**	トリガーをアクティブにします。

次の作業

上述の設定が機能することを確認するために、インターフェイスの少なくとも 1 つに 1 分間で 30 パケットより多くのパケットを取得させます。これにより、1 分後にトラップが送信されるはずです。

トリガーのアクティブ化

トリガーをアクティブにするには、次の手順を実行します。

手順の概要

1. **setany-v2c$ADDRESSprivatemteTriggerEntryStatus.4.106.111.104.110.1-i1**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**setany-v2c$ADDRESSprivatemteTriggerEntryStatus.4.106.111.104.110.1-i1**	トリガーをアクティブにします。

次の作業

上述の設定が機能することを確認するために、インターフェイスの少なくとも 1 つに 1 分間で 30 パケットより多くのパケットを取得させます。これにより、1 分後にトラップが送信されるはずです。

イベント MIB のモニタリングおよびメンテナンス

イベント MIB のアクティビティを Cisco コマンドライン インターフェイスからモニタするには、次のコマンドを使用します。

コマンド	目的
debug management event mib	イベント MIB が指定されたトリガーを評価するたびに画面にメッセージを出力します。これらのメッセージは、テクニカル サポートのエンジニアがトラブルシューティング目的で使用することを想定したもので、リアルタイムで与えられます。
show management event	イベント MIB の使用を通じてルーティング デバイス上で設定された SNMP イベント値を表示します。

コマンドライン インターフェイスを使用したイベント MIB の設定

イベント MIB は、SNMP を使用して直接設定できます。この手順では、1 分おきにすべてのインターフェイスの ifInOctets のデルタ値をモニタするようにイベント MIB を設定します。指定したしきい値を超えるサンプルがあった場合、トラップ通知が送信されます。

リリースによっては、イベントMIB機能が拡張され、イベント、イベントアクション、およびトリガーを設定するコマンドラインインターフェイスコマンドが追加されていることに注意してください。

ここでは、イベント MIB を設定する手順について説明します。

スカラ変数の設定

イベント MIB のスカラ変数を設定するには、次の手順を実行します。

はじめる前に

イベント MIB のスカラ変数を設定するには、イベント MIB スカラ変数に慣れている必要があります。

手順の概要

1. **イネーブル化**
2. **configureterminal**
3. **snmpmibeventsampleminimum***value*
4. **snmpmibeventsampleinstancemaximum***value*
5. **exit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**イネーブル化** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	**snmpmibeventsampleminimum***value* 例： `Device(config)# snmp mib event sample minimum 10`	オブジェクトサンプリングの最小値を設定します。
ステップ 4	**snmpmibeventsampleinstancemaximum***value* 例： `Device(config)# snmp mib event sample instance maximum 50`	オブジェクトインスタンス サンプリングの最大値を設定します。
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 モードを終了します。

イベント MIB オブジェクト リストの設定

イベントMIBを設定するには、トリガー、トリガーテスト、またはイベントに従って通知に追加できるオブジェクトのリストをセットアップする必要があります。

はじめる前に

イベント MIB オブジェクト リストを設定するには、イベント MIB オブジェクトとオブジェクト ID に慣れている必要があります。これらは、イベント、トリガー、またはトリガーテストに従って通知に追加できるものです。

手順の概要

1. イネーブル化
2. **configureterminal**
3. **snmpmibeventobjectlistowner***object-list-owner***name***object-list-nameobject-number*
4. **objectid***object-identifier*
5. **wildcard**
6. **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	イネーブル化 例： `Device> enable`	特権EXECモードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバルコンフィギュレーションモードを開始します。
ステップ 3	**snmpmibeventobjectlistowner***object-list-owner***name***object-list-nameobject-number* 例： `Device(config)# snmp mib event object list owner owner1 name objectA 10`	イベント MIB オブジェクトリストを設定します。
ステップ 4	**objectid***object-identifier* 例： `Device(config-event-objlist)# object id ifInOctets`	イベントに設定されるオブジェクトのオブジェクト ID を指定します。
ステップ 5	**wildcard** 例： `Device(config-event-objlist)# wildcard`	（任意）オブジェクト ID に、ワイルドカードを使用した検索を開始します。オブジェクト ID を部分的に指定することにより、オブジェクト ID のリストを取得できます。
ステップ 6	**end** 例： `Device(config-event-objlist)# end`	オブジェクトリストコンフィギュレーションモードを終了します。

イベントの設定

管理イベントを設定するには、この手順を実行します。

はじめる前に

管理イベントを設定するには、SNMP MIB のイベントとオブジェクト ID に慣れている必要があります。

手順の概要

1. イネーブル化
2. **configureterminal**
3. **snmpmibeventowner***event-owner***name***event-name*
4. **description***event-description*
5. イネーブル化
6. **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	イネーブル化 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**snmpmibeventowner***event-owner***name***event-name* 例： `Device(config)# snmp mib event owner owner1 name EventA`	イベント 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***event-description* 例： `Device(config-event)# description "EventA is an RMON event"`	イベントの機能と用途を説明します。
ステップ 5	イネーブル化 例： `Device(config-event)# enable`	イベントをイネーブルにします。 （注） イベント トリガー中にイベントが実行できるのは、それがイネーブルになっている場合だけです。

	コマンドまたはアクション	目的
ステップ 6	**end** 例： `Device(config-event)# end`	イベント コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

イベント アクションの設定

イベントアクションを設定することにより、イベントトリガー中にアプリケーションが実行できるアクションを定義できます。イベントのアクションには、通知の送信やMIBオブジェクトの送信などがあります。イベント アクション情報は **set** または **notification** に設定できます。このイベントのアクションは、イベント コンフィギュレーション モードのみで設定できます。

ここでは、イベント アクションを設定するための次の手順を説明します。

アクション通知の設定

イベントの通知アクションを設定するには、次の手順を実行します。

手順の概要

1. イネーブル化
2. **configure terminal**
3. **snmp mib event owner** *event-owner* **name** *event-name*
4. **actionnotification**
5. **objectid***object-id*
6. **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	イネーブル化 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
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 mib event owner** *event-owner* **name** *event-name* 例： `Device(config)# snmp mib event owner owner1 event EventA`	イベント コンフィギュレーション モードになります。
ステップ 4	**actionnotification** 例： `Device(config-event)# action notification`	イベントの通知アクションを設定します。 （注） イベント アクションを notification に設定すると、イベントに関連付けられているオブジェクトに変更が加えられるたびに通知が生成されます。
ステップ 5	**objectid***object-id* 例： `Device(config-event-action-notification)# object id ifInOctets`	アクション通知のオブジェクトを設定します。指定されたオブジェクトが変更されると、ホスト システムに通知が送信されます。
ステップ 6	**end** 例： `Device(config-event-action-notification)# end`	アクション通知コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

アクション セットの設定

イベントのアクションを設定するには、この手順を実行します。

手順の概要

1. イネーブル化
2. **configure terminal**
3. **actionset**
4. **objectid***object-id*
5. **value***integer-value*
6. **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	イネーブル化	特権 EXEC モードをイネーブルにします。

	コマンドまたはアクション	目的
	例： `Device> enable`	• パスワードを入力します（要求された場合）。
ステップ 2	**configure 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	**actionset** 例： `Device(config-event)# action set`	アクションセット 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	**objectid***object-id* 例： `Device(config-event-action-set)# object id ifInOctets`	アクション セットのオブジェクトを設定します。指定したオブジェクトに変更が加えられると、指定したアクションが実行されます。
ステップ 5	**value***integer-value* 例： `Device(config-event-action-set)# value 10`	オブジェクトの値を設定します。
ステップ 6	**end** 例： `Device(config-event-action-set)# end`	アクションセット コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

イベント トリガーの設定

イベント トリガーを設定することにより、モニタするオブジェクトをリストし、各トリガーをイベントに関連付けることができます。

イベント トリガーを設定するには、次の手順を実行します。

手順の概要

1. イネーブル化
2. **configureterminal**
3. **snmpmibeventtriggerowner***trigger-owner***name***trigger-name*
4. **description***trigger-description*
5. **frequency***seconds*
6. **objectlistowner***object-list-owner***name***object-list-name*
7. **objectid***object-identifier*
8. イネーブル化
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	イネーブル化 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**snmpmibeventtriggerowner***trigger-owner***name***trigger-name* 例： `Device(config)# snmp mib event trigger owner owner1 name EventTriggerA`	指定されたイベント トリガーのイベント トリガー コンフィギュレーション モードをイネーブルにします。
ステップ 4	**description***trigger-description* 例： `Device(config-event-trigger)# description "EventTriggerA is an RMON alarm."`	イベントトリガーの機能と用途を説明します。
ステップ 5	**frequency***seconds* 例： `Device(config-event-trigger)# frequency 120`	トリガーサンプル間の待機時間（秒単位）を設定します。

	コマンドまたはアクション	目的
ステップ 6	**objectlistowner***object-list-owner***name***object-list-name* 例： `Device(config-event-trigger)# object list owner owner1 name ObjectListA`	通知に追加できるオブジェクトのリストを指定します。
ステップ 7	**objectid***object-identifier* 例： `Device(config-event-trigger)# object id ifInOctets`	イベント トリガーのオブジェクト ID を設定します。
ステップ 8	イネーブル化 例： `Device(config-event-trigger)# enable`	イベントトリガーをイネーブルにします。
ステップ 9	**end** 例： `Device(config-event-trigger)# end`	イベント トリガー コンフィギュレーション モードを終了します。

存在トリガー テストの設定

このトリガー タイプの設定は、イベント トリガー コンフィギュレーション モードで行う必要があります。

テスト存在トリガー タイプのためのトリガー パラメータを設定するには、次の手順を実行します。

手順の概要

1. **testexistence**
2. **eventowner***event-owner***name***event-name*
3. **objectlistowner***object-list-owner***name***object-list-name*
4. **type**{**present** | **absent** | **changed**}
5. **startup**{**present** | **absent**}
6. **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	**testexistence** 例： `Device(config-event-trigger)# test existence`	テスト存在コンフィギュレーションモードをイネーブルにします。
ステップ 2	**eventowner***event-owner***name***event-name* 例： `Device(config-event-trigger-existence)# event owner owner1 name EventA`	存在トリガー テストのイベントを設定します。
ステップ 3	**objectlistowner***object-list-owner***name***object-list-name* 例： `Device(config-event-trigger-existence)# object list owner owner1 name ObjectListA`	存在トリガーテストのためのオブジェクトのリストを設定します。
ステップ 4	**type**{**present** | **absent** | **changed**} 例： `Device(config-event-trigger-existence)# type present`	指定したタイプの存在テストを実行します。 存在テストは次の 3 種類があります。 • present：イベント トリガーが存続している間にオブジェクトが現れる場合は、タイプを present テストに設定します。 • absent：イベント トリガーが存続している間にオブジェクトが消える場合は、タイプを absent テストに設定します。 • changed：イベント トリガーが存続している間にオブジェクトに変更が加えられる場合は、タイプを changed テストに設定します。
ステップ 5	**startup**{**present** | **absent**} 例： `Device(config-event-trigger-existence)# startup present`	テストが正常に実行された場合にイベントをトリガーします。
ステップ 6	**end** 例： `Device(config-event-trigger-existence)# end`	存在トリガー テスト コンフィギュレーション モードを終了します。

ブール値トリガー テストの設定

このトリガー テストの設定は、イベント トリガー コンフィギュレーション モードで行う必要があります。

ブール値トリガータイプのためのトリガーパラメータを設定するには、次の手順を実行します。

手順の概要

1. **testboolean**
2. **comparison**{**unequal** | **equal** | **less** | **lessOrEqual** | **greater** | **greaterOrEqual**}
3. **value***integer-value*
4. **objectlistowner***object-list-owner***name***object-list-name*
5. **eventowner***event-owner***name***event-name*
6. **startup**
7. **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	**testboolean** 例： `Device(config-event-trigger)# test boolean`	ブール値トリガー テスト コンフィギュレーション モードをイネーブルにします。
ステップ 2	**comparison**{**unequal** | **equal** | **less** | **lessOrEqual** | **greater** | **greaterOrEqual**} 例： `Device(config-event-trigger-boolean)# comparison unequal`	指定されたブール値比較テストを実行します。 • ブール値比較テストの値には、unequal、equal、less、lessOrEqual、greater、greaterOrEqual のいずれかを設定できます。
ステップ 3	**value***integer-value* 例： `Device(config-event-trigger-boolean)# value 10`	ブール値トリガー テストの値を設定します。
ステップ 4	**objectlistowner***object-list-owner***name***object-list-name* 例： `Device(config-event-trigger-boolean)# object list owner owner1 name ObjectListA`	ブール値トリガー テストのためのオブジェクトのリストを設定します。
ステップ 5	**eventowner***event-owner***name***event-name* 例： `Device(config-event-trigger-boolean)# event owner owner1 name EventA`	ブール値トリガー タイプのイベントを設定します。

	コマンドまたはアクション	目的
ステップ 6	**startup** 例： `Device(config-event-trigger-boolean)# startup`	テストが正常に実行された場合にイベントをトリガーします。
ステップ 7	**end** 例： `Device(config-event-trigger-boolean)# end`	ブール値トリガー テスト コンフィギュレーション モードを終了します。

しきい値トリガー テストの設定

このトリガー テストの設定は、イベント トリガー コンフィギュレーション モードで行う必要があります。

しきい値トリガー テストのトリガー パラメータを設定するには、次の手順を実行します。

手順の概要

1. **testthreshold**
2. **objectlistowner***object-list-owner***name***object-list-name*
3. **rising***integer-value*
4. **risingeventowner***event-owner***name***event-name*
5. **falling***integer-value*
6. **fallingeventowner***event-owner***name***event-name*
7. **deltarising***integer-value*
8. **deltarisingeventowner***event-owner***name***event-name*
9. **deltafalling***integer-value*
10. **deltafallingeventowner***event-owner***name***event-name*
11. **startup**{**rising** | **falling** | **rising-or-falling**}
12. **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	**testthreshold** 例： `Device(config-event-trigger)# test threshold`	しきい値トリガー テスト コンフィギュレーション モード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 2	**objectlistowner***object-list-owner***name***object-list-name* 例： `Device(config-event-trigger-threshold)# object list owner owner1 name ObjectListA`	しきい値トリガー テストのためのオブジェクトのリストを設定します。
ステップ 3	**rising***integer-value* 例： `Device(config-event-trigger-threshold)# rising 100`	上昇しきい値を指定された値に設定します。
ステップ 4	**risingeventowner***event-owner***name***event-name* 例： `Device(config-event-trigger-threshold)# rising event owner owner1 name EventA`	上昇しきい値のしきい値トリガー テストのイベントを設定します。
ステップ 5	**falling***integer-value* 例： `Device(config-event-trigger-threshold)# falling 50`	下限しきい値を指定された値に設定します。
ステップ 6	**fallingeventowner***event-owner***name***event-name* 例： `Device(config-event-trigger-threshold)# falling event owner owner1 name EventB`	下限しきい値のしきい値トリガー テストのイベントを設定します。
ステップ 7	**deltarising***integer-value* 例： `Device(config-event-trigger-threshold)# delta rising 30`	イベント トリガーに指定されているサンプリング方式がデルタの場合に、デルタ上昇しきい値を指定された値に設定します。
ステップ 8	**deltarisingeventowner***event-owner***name***event-name* 例： `Device(config-event-trigger-threshold)# delta rising event owner owner1 name EventC`	デルタ上昇しきい値のしきい値トリガー テストのイベントを設定します。
ステップ 9	**deltafalling***integer-value* 例： `Device(config-event-trigger-threshold)# delta falling 10`	イベント トリガーに指定されているサンプリング方式がデルタの場合に、デルタ下限しきい値を指定された値に設定します。

	コマンドまたはアクション	目的
ステップ 10	**deltafallingeventowner***event-owner***name***event-name* 例： `Device(config-event-trigger-threshold)# delta falling event owner owner1 name EventAA`	デルタ下限しきい値のしきい値トリガー テストのイベントを設定します。
ステップ 11	**startup**{**rising** | **falling** | **rising-or-falling**} 例： `Device(config-event-trigger-threshold)# startup rising`	しきい値トリガー テストの条件が満たされた場合に、イベントをトリガーします。
ステップ 12	**end** 例： `Device(config-event-trigger-threshold)# end`	しきい値トリガーテスト コンフィギュレーション モードを終了します。

SNMP を使用する式 MIB の設定

式 MIB は、SNMP を使用して直接設定できます。

式 MIB に関連付けられた Cisco ソフトウェア設定手順はありません。式 MIB 機能の設定はすべて、SNMP を使用して、アプリケーションを通じて実行する必要があります。ここでは、外部デバイス上のネットワーク管理アプリケーションを使用したサンプル設定セッションを示します。お使いの Cisco ルーティング デバイスでの SNMP の設定については、「その他の参考資料」セクションを参照してください。

以降では、Sun のワークステーションで使用できる SNMP 調査ツールを使用した式 MIB の設定方法をステップ バイ ステップで示します。ここで示す **setany** コマンドは、SNMP アプリケーションを使用して実行されます。これらのコマンドは Cisco コマンドライン インターフェイス コマンドではないことに注意してください。使用しているルーティング デバイス上で SNMP が設定されていることを前提としています。

次の設定では、カウンタ ifInOctects と ifOutOctects の追加を含むワイルドカード式が評価されます。

手順の概要

1. **setany-v2c$SNMP_HOSTprivateexpResourceDeltaMinimum.0-i60**
2. **setany-v2c$SNMP_HOSTprivateexpExpressionIndex.116.101.115.116-g9**
3. **setany-v2c$SNMP_HOSTprivateexpNameStatus.116.101.115.116-i5**
4. **setany-v2c$SNMP_HOSTprivateexpExpressionComment.9-D"testexpression"**
5. **setany-v2c$SNMP_HOSTprivateexpExpression.9-D'$1+$2'**
6. **setany-v2c$SNMP_HOSTprivateexpObjectID.9.1-difInOctets**
7. **setany-v2c$SNMP_HOSTprivateexpObjectSampleType.9.1-i2**
8. **setany-v2c$SNMP_HOSTprivateexpObjectIDWildcard.9.1-i1**
9. **setany-v2c$SNMP_HOSTprivateexpObjectStatus.9.1-i1**
10. **setany-v2c$SNMP_HOSTprivateexpNameStatus.116.101.115.116-i1**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**setany-v2c$SNMP_HOSTprivateexpResourceDeltaMinimum.0-i60**	システムが受け付ける最小デルタ間隔を設定します。
ステップ 2	**setany-v2c$SNMP_HOSTprivateexpExpressionIndex.116.101.115.116-g9**	式の識別に使用される ID 番号を設定します。 • expName は、たとえば、「test」（ASCII で 116.101.115.116）にすることができます。
ステップ 3	**setany-v2c$SNMP_HOSTprivateexpNameStatus.116.101.115.116-i5**	expNameStatusTable 内にエントリを作成します。 （注） expNameTable でエントリが作成される場合、これによって expExpressionTable 内にエントリが自動的に作成されます。
ステップ 4	**setany-v2c$SNMP_HOSTprivateexpExpressionComment.9-D"testexpression"**	式の用途または意味を説明するコメントへのオブジェクトを設定します。 • ここでは、コメントは「test expression」です。

	コマンドまたはアクション	目的
ステップ 5	**setany-v2c$SNMP_HOSTprivateexpExpression.9-D'$1+$2'**	オブジェクト expExpression に、評価する必要のある式を設定します。 • この式は「$1」が「ifInOctets」、「$2」が「ifOutOctets」に対応し、2 台のカウンタ オブジェクトの追加を示します。
ステップ 6	**setany-v2c$SNMP_HOSTprivateexpObjectID.9.1-difInOctets** 例： `setany -v2c $SNMP_HOST private expObjectID.9.2 -d ifOutOctets`	前述の式で使用されるオブジェクト ID を計算のために指定します。 • ここで、オブジェクト expObjectID の末尾に付いている番号「9」は式を識別する一意の識別子です。また、「9」の次の番号「1」は式内のオブジェクトを識別する別の一意の識別子です。式の作成に使用されている 2 つのオブジェクトに expObjectID を設定します。
ステップ 7	**setany-v2c$SNMP_HOSTprivateexpObjectSampleType.9.1-i2** 例： `setany -v2c $SNMP_HOST private expObjectSampleType.9.2 -i 2`	式内のオブジェクトに対して行われるサンプリングのタイプを設定します。 • サンプリングには、a) 絶対（absolute）b) デルタ（delta）の 2 つのタイプがあります。ここでは、サンプリングのタイプは「デルタ」に設定されています。
ステップ 8	**setany-v2c$SNMP_HOSTprivateexpObjectIDWildcard.9.1-i1** 例： `setany -v2c $SNMP_HOST private expObjectIDWildcard.9.2 -i 1`	expObjectID にワイルドカードが使用されるかどうかを指定します。ここでは、両方の expObjectID がワイルドカードになります。

	コマンドまたはアクション	目的
ステップ 9	**setany-v2c$SNMP_HOSTprivateexpObjectStatus.9.1-i1** 例： `setany -v2c $SNMP_HOST private expObjectStatus.9.2 -i 1`	expObjectTable 内の行がアクティブになるように設定します。
ステップ 10	**setany-v2c$SNMP_HOSTprivateexpNameStatus.116.101.115.116-i1**	expNameTable 内の行をアクティブに設定すると、式の値が評価できます。 • 式の値が expValueTable から取得できるようになりました。

CLI を使用した式 MIB の設定

式 MIB は、SNMP を使用して直接設定できます。ただし、Cisco IOS Release 12.4(20)T では、式 MIB の機能が強化され、式を設定するための CLI が追加されました。式 MIB を設定する前に、式、オブジェクト ID、およびサンプリング方式に慣れている必要があります。

ここでは、式 MIB を設定するための次の手順を説明します。

式 MIB のスカラ オブジェクトの設定

式 MIB は、次のスカラ オブジェクトを持ちます。

- expResourceDeltaMinimum
- expResourceDeltaWildcardInstanceMaximum

式 MIB のスカラ オブジェクトを設定するには、次の手順を実行します。

手順の概要

1. イネーブル化
2. **configureterminal**
3. **snmpmibexpressiondeltaminimum***seconds*
4. **snmpmibexpressiondeltawildcardmaximum***number-of-instances*
5. **exit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	イネーブル化 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**snmpmibexpressiondeltaminimum***seconds* 例： `Device(config)# snmp mib expression delta minimum 20`	（任意）最小デルタ間隔を秒単位で設定します。 （注） アプリケーションは、定常的にデルタを計算することによる影響を低減するためにこの最小デルタ間隔にもっと大きい値を使用する場合があります。デルタ サンプリング間隔が大きくなると、アプリケーションがサンプリングを行う頻度が下がり、オーバーヘッドが少なくなります。このコマンドを使用することにより、デルタ間隔を設定した後に作成したすべての式のオーバーヘッド低減を強制できます。
ステップ 4	**snmpmibexpressiondeltawildcardmaximum***number-of-instances* 例： `Device(config)# snmp mib expression delta wildcard maximum 120`	（任意）式内のワイルドカードが使用されたデルタ オブジェクトのダイナミック インスタンス エントリの最大数を制限します。 与えられたデルタ式で、ダイナミック インスタンスの数は、存在するすべての条件を満たす値の数に、式内のデルタ値の数を掛けたものです。インスタンス エントリには、あらかじめ設定された制限はなく、この値はシステムのリソースに基づいてダイナミックに変動します。
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 モードを終了し、特権 EXEC モードに戻ります。

式の設定

式を設定するには、次の手順を実行します。

手順の概要

1. イネーブル化
2. **configureterminal**
3. **snmpmibexpressionowner***expression-owner***name***expression-name*
4. **description***expression-description*
5. **expression***expression*
6. **deltainterval***seconds*
7. **valuetype**{**counter32** | **unsigned32** | **timeticks** | **integer32** | **ipaddress** | **octetstring** | **objectid** | **counter64**}
8. イネーブル化
9. **object***object-number*
10. **id***object-identifier*
11. **wildcard**
12. **discontinuityobject***discontinuity-object-id*[**wildcard**] [**type** {**timeticks** | **timestamp** | **date-and-time**}]
13. **conditionalobject***conditional-object-id***[wildcard]**
14. **sample**{**absolute** | **delta** | **changed**}
15. **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	イネーブル化 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**snmpmibexpressionowner***expression-owner***name***expression-name* 例： `Device(config-expression)# snmp mib expression owner owner1 name ExpA`	設定する式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 4	**description***expression-description* 例： `Device(config-expression)# description this expression is created for the sysLocation MIB object`	式の説明を設定します。
ステップ 5	**expression***expression* 例： `Device(config-expression)# expression ($1+$2)*800/$3`	式が評価されるように設定します。 （注） この式は、ANSI C の構文です。ただし、式内の変数は、ドル記号（$）と式の評価に使用されるオブジェクトのオブジェクト番号を表す整数値の組み合わせとして定義されます。
ステップ 6	**deltainterval***seconds* 例： `Device(config-expression)# delta interval 180`	サンプリング方式がデルタの場合は、式内のオブジェクトのサンプリング間隔を設定します。
ステップ 7	**valuetype**{**counter32** | **unsigned32** | **timeticks** | **integer32** | **ipaddress** | **octetstring** | **objectid** | **counter64**} 例： `Device(config-expression)# value type counter32`	式に指定する値の型を設定します。
ステップ 8	イネーブル化 例： `Device(config-expression)# enable`	評価する式をイネーブルにします。
ステップ 9	**object***object-number* 例： `Device(config-expression)# object 2`	式の評価に使用されるオブジェクトを設定します。 • オブジェクトを式内の変数と関連付けるために、オブジェクト番号が使用されます。オブジェクトに対応する変数は、$ とオブジェクト番号です。従って、ここで使用した例内の変数は、$10 に対応します。
ステップ 10	**id***object-identifier* 例： `Device(config-expression-object)# id ifInOctets`	オブジェクト ID を設定します。

	コマンドまたはアクション	目的
ステップ 11	**wildcard** 例： `Device(config-expression-object)# wildcard`	（任意）式の評価に使用されるオブジェクトのワイルドカード検索をイネーブルにします。
ステップ 12	**discontinuityobject***discontinuity-object-id***[wildcard]** [**type** {**timeticks** | **timestamp** | **date-and-time**}] 例： `Device(config-expression-object)# discontinuity object sysUpTime`	（任意）オブジェクトサンプリングタイプがデルタに設定されたか、変更された場合、オブジェクトの不連続プロパティを設定します。不連続オブジェクトIDでは、カウンタの不連続に対する通常のチェックがサポートされます。 • **wildcard** キーワードを使用することにより、不連続プロパティでのオブジェクトのワイルドカード検索をイネーブルにできます。 • **type** キーワードを使用することにより、オブジェクトに不連続プロパティの値を設定できます。
ステップ 13	**conditionalobject***conditional-object-id***[wildcard]** 例： `Device(config-expression-object)# conditional object mib-2.90.1.3.1.1.2.3.112.99.110.4.101.120.112.53`	（任意）条件オブジェクト ID を設定します。 • **wildcard** キーワードを使用することにより、不連続プロパティでの条件オブジェクトのワイルドカード検索をイネーブルにできます。
ステップ 14	**sample**{**absolute** | **delta** | **changed**} 例： `Device(config-expression-object)# sample delta`	オブジェクトに対して、指定されているサンプリング方式をイネーブルにします。この例では、デルタサンプリング方式を使用しています。 絶対値、デルタ、変更の 3 つのサンプリング方式のいずれかを設定できます。 • 絶対値サンプリング：サンプリング中に MIB オブジェクトの値を使用します。 • デルタサンプリング：アプリケーションで維持されている最後のサンプリング値を使用します。この方式には、アプリケーションが連続的にサンプリングを行うことが必要です。

	コマンドまたはアクション	目的
		• 変更サンプリング：最後のサンプル以降に変更されたオブジェクトの値を使用します。
ステップ 15	**end** 例： `Device(config-expression-object)# end`	式オブジェクト コンフィギュレーション モードを終了します。

SNMP サポートの設定例

例：SNMPv1、SNMPv2c、および SNMPv3 の設定

次の例は、SNMPv1、SNMPv2c、および SNMPv3 をイネーブルにする方法を示します。この設定は、任意の SNMP マネージャが、public という名前のコミュニティ ストリングを使用してすべてのオブジェクトに読み取り専用の権限でアクセスできるようにします。この設定は、デバイスによるトラップの送信は発生させません。

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

次の例は、すべてのオブジェクトへの public という名前のコミュニティ ストリングを使用した読み取り専用権限での SNMP アクセスを許可する方法を示します。デバイスが ISDN トラップを、SNMPv1 を使用してホスト 172.16.1.111 および 172.16.1.33 に、SNMPv2c を使用してホスト 172.16.1.27 に送信します。トラップとともにコミュニティ ストリング public が送信されます。

```
Device(config)# snmp-server community public
Device(config)# snmp-server enable traps isdn
Device(config)# snmp-server host 172.16.1.27 version 2c public
Device(config)# snmp-server host 172.16.1.111 version 1 public
Device(config)# snmp-server host 172.16.1.33 public
```

次に、comaccess コミュニティ ストリングを指定するアクセス リスト 4 のメンバに対してすべてのオブジェクトへの読み取り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 認証エラー トラップが、public というコミュニティ ストリングを使用して、SNMPv2c によってホスト example.com に送信されます。

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

次の例は、SNMPv3 セキュリティ モデルがイネーブルになっている場合に、noAuthNoPriv セキュリティ レベルでトラップを受信するようにリモート ユーザを設定する方法を示します。

```
Device(config)# snmp-server group group1 v3 noauth
```

```
Device(config)# snmp-server user remoteuser1 group1 remote 10.12.8.4
Device(config)# snmp-server host 10.12.8.4 informs version 3 noauth remoteuser config
```

次の例は、SNMPv3 セキュリティ モデルがイネーブルになっている場合に、authNoPriv セキュリティ レベルでトラップを受信するようにリモート ユーザを設定する方法を示します。

```
Device(config)# snmp-server group group2 v3 auth
Device(config)# snmp-server user AuthUser group2 remote 10.12.8.4 v3 auth md5 password1
```

次の例は、SNMPv3 セキュリティ モデルがイネーブルになっている場合に、priv セキュリティ レベルでトラップを受信するようにリモート ユーザを設定する方法を示します。

```
Device(config)# snmp-server group group3 v3 priv
Device(config)# snmp-server user PrivateUser group3 remote 10.12.8.4 v3 auth md5 password1
 priv access des56
```

次の例は、エンティティ MIB インフォーム通知をホスト example.com に送信する方法を示します。コミュニティ ストリングは制限されます。最初の行は、これまでにイネーブルになっていたトラップやインフォームに加えて、デバイスによるエンティティ MIB 通知の送信をイネーブルにします。2 行目は、通知がインフォームとして送信されるように指定し、これらのインフォームの宛先を指定し、ホスト example.com に対するこれまでの **snmp-serverhost** コマンドを上書きします。

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

次の例は、SNMP および Cisco 環境モニタ エンタープライズ専用トラップをアドレス 172.30.2.160 に送信する方法を示します。

```
Device(config)# snmp-server enable traps
Device(config)# snmp-server host 172.30.2.160 public snmp envmon
```

次の例は、デバイスによる、コミュニティ ストリング public を使用した、ホスト example.com へのすべてのトラップの送信をイネーブルにする方法を示します。

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

次の例は、トラップが一切ホストに送信されない設定を示します。BGP トラップはすべてのホストに対してイネーブルになっていますが、ISDN トラップは 1 つのホストに送信されるようにイネーブルになっています。

```
Device(config)# snmp-server enable traps bgp
Device(config)# snmp-server host host1 public isdn
```

次の例は、デバイスによる、public というコミュニティ ストリングを使用した、ホスト example.com へのすべてのインフォームの送信をイネーブルにする方法を示します。

```
Device(config)# snmp-server enable traps
Device(config)# snmp-server host example.com informs version 2c public
```

次の例では、SNMP マネージャをイネーブルにし、セッション タイムアウトをデフォルトより大きい値に設定しています。

```
Device(config)# snmp-server manager
Device(config)# snmp-server manager session-timeout 1000
```

例：ifAlias ロング ネーム サポートの設定

次の例では、長い説明をスロット 1、ポート アダプタ 0、ポート 0 にあるファスト イーサネット インターフェイスに適用しています。

```
Device# configure terminal
Device(config)#interface FastEthernet1/0/0
Device(config-if)# description FastEthernet1/0/0 this is a test of a description that exceeds
 64 characters in length
Device(config-if)#ip address 192.168.134.55 255.255.255.0
Device(config-if)#no ip directed-broadcast
Device(config-if)#no ip route-cache distributed
```

次の例は、ifAlias ロング ネーム サポートがまだイネーブルになっていない（デフォルト）ことを前提として、NMS からの mibwalk 操作の結果を示します。

```
***** SNMP QUERY STARTED *****
 .
 .
 .
 ifXEntry.18.10 (octets) (zero-length)
 ifXEntry.18.11 (octets) Fastethernet1/0/0 this is a test of a description that exceeds 64
 ch
 ifXEntry.18.12 (octets) (zero-length)
 .
 .
 .
```

次の出力は、CLI に表示される説明を示します。

```
Device# show interface FastEthernet0/0/0

FastEthernet1/0/0 is administratively down, line protocol is down
  Hardware is Lance, address is 0010.7b4d.7046 (bia 0010.7b4d.7046)
  Description: FastEthernet1/0/0 this is a test of a description that exceeds 64 chh
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec,
     reliability 252/255, txload 1/255, rxload 1/255
 .
 .
 .
```

次の例では、ifAlias ロング ネーム サポートをイネーブルにし、再び説明を表示します。

```
Device(config)# snmp ifmib ifalias long
Device(config)#interface FastEthernet1/0/0
Device(config-if)# description FastEthernet1/0/0 this is a test of a description that exceeds
64 characters in length
Device(config)#end

Device# show interface FastEthernet1/0/0

FastEthernet1/0/0 is administratively down, line protocol is down
  Hardware is Lance, address is 0010.7b4d.7046 (bia 0010.7b4d.7046)
  Description: FastEthernet1/0/0 this is a test of a description that exceeds 64 characters
 in length
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec,
     reliability 252/255, txload 1/255, rxload 1/255
 .
 .
 .
***** SNMP QUERY STARTED *****
 .
 .
 .
```

```
ifXEntry.18.10 (octets) (zero-length)
ifXEntry.18.11 (octets) FastEthernet1/0/0 this is a test of a description that exceeds 64
characters in length
ifXEntry.18.12 (octets) (zero-length)
.
.
.
```

例：VPN のための SNMP サポートの設定

次に、すべての SNMP 通知が trap-vrf という名前の VRF を介して example.com に送信される例を示します。

```
Device(config)# snmp-server host example.com vrf trap-vrf
```

次に、「traps-vrf」という名前の VRF がリモート サーバ 172.16.20.3 に対して設定される例を示します。

```
Device(config)# snmp-server engineID remote 172.16.20.3 vrf traps-vrf 80000009030000B064EFE100
```

例：イベント MIB の設定

次の例は、イベントのスカラ変数を設定する方法を示します。

```
Device# configure terminal
Device(config)# snmp mib event sample minimum 10
Device(config)# snmp mib event sample instance maximum 50
Device(config)# exit
```

次の例は、イベントのオブジェクト リストを設定する方法を示します。

```
Device# configure terminal
Device(config)# snmp mib event object list owner owner1 name objectA 1
Device(config-event-objlist)# object id ifInOctets
Device(config-event-objlist)# wildcard
Device(config-event-objlist)# exit
```

次の例は、イベントを設定する方法を示します。

```
Device# configure terminal
Device(config)# snmp mib event owner owner1 name EventA
Device(config-event)# description "eventA is an RMON event."
Device(config-event)# enable
Device(config-event)# exit
```

次の例は、イベントの通知アクションを設定する方法を示します。

```
Device(config-event)# action notification
Device(config-event-action-notification)# object id ifInOctets
Device(config-event-action-notification)# exit
```

次の例は、イベントのアクションを設定する方法を示します。

```
Device(config-event)# action set
Device(config-event-action-set)# object id ifInOctets
Device(config-event-action-set)# value 10
Device(config-event-action-set)# exit
```

次の例は、イベントのトリガーを設定する方法を示します。

```
Device# configure terminal
Device(config)# snmp mib event trigger owner owner1 name EventTriggerA
Device(config-event-trigger)# description “EventTriggerA is an RMON alarm.”
Device(config-event-trigger)# frequency 120
Device(config-event-trigger)# object list owner owner1 name ObjectListA
```

```
Device(config-event-trigger)# object id ifInOctets
Device(config-event-trigger-object-id)# enable
Device(config-event-trigger)# exit
```

次の例は、存在トリガー テストを設定する方法を示します。

```
Device(config-event-trigger)# test existence
Device(config-event-trigger-existence)# event owner owner1 name EventA
Device(config-event-trigger-existence)# object list owner owner1 name ObjectListA
Device(config-event-trigger-existence)# type present
Device(config-event-trigger-existence)# startup present
Device(config-event-trigger-existence)# exit
```

次の例は、ブール値トリガー テストを設定する方法を示します。

```
Device(config-event-trigger)# test boolean
Device(config-event-trigger-boolean)# comparison unequal
Device(config-event-trigger-boolean)# value 10
Device(config-event-trigger-boolean)# object list owner owner1 name ObjectListA
Device(config-event-trigger-boolean)# event owner owner1 name EventA
Device(config-event-trigger-boolean)# startup
Device(config-event-trigger-boolean)# exit
```

次の例は、しきい値トリガー テストを設定する方法を示します。

```
Device(config-event-trigger)# test threshold
Device(config-event-trigger-threshold)# object list owner owner1 name ObjectListA
Device(config-event-trigger-threshold)# rising 100
Device(config-event-trigger-threshold)# rising event owner owner1 name EventA
Device(config-event-trigger-threshold)# falling 50
Device(config-event-trigger-threshold)# falling event owner owner1 name EventA
Device(config-event-trigger-threshold)# delta rising 30
Device(config-event-trigger-threshold)# delta rising event owner owner1 name EventA
Device(config-event-trigger-threshold)# delta falling 10
Device(config-event-trigger-threshold)# delta falling event owner owner1 name EventA
Device(config-event-trigger-threshold)# startup rising
Device(config-event-trigger-threshold)# exit
```

例：式 MIB の設定

次の例は、グローバル コンフィギュレーション モードで **snmpmibexpression** コマンドを使用して式 MIB を設定にする方法を示します。

```
Device(config)# snmp mib expression owner pcn name exp6
Device(config-expression)# description this expression is created for the sysLocation MIB object
Device(config-expression)# expression ($1+$2)*800/$3
Device(config-expression)# delta interval 120
Device(config-expression)# value type counter32
Device(config-expression)# enable
Device(config-expression)# object 2
Device(config-expression-object)# id ifInOctets
Device(config-expression-object)# wildcard
Device(config-expression-object)# discontinuity object sysUpTime
Device(config-expression-object)# conditional object mib-2.90.1.3.1.1.2.3.112.99.110.4.101.120.112.53
 wildcard
Device(config-expression-object)# sample delta
Device(config-expression-object)# end
```

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
Cisco IOS コマンド	『Cisco IOS Master Commands List, All Releases』
『Cisco IOS SNMP Support Command Reference』	『Cisco IOS SNMP Support Command Reference』

標準および RFC

標準/RFC	Title
CBC-DES（DES-56）標準	『*Symmetric Encryption Protocol*』

標準/RFC	Title
規格 58	『*Structure of Management Information Version 2 (SMIv2)*』>
RFC 1067	『*A Simple Network Management Protocol*』
RFC 1091	『*Telnet terminal-type option*』
RFC 1098	*Simple Network Management Protocol*（*SNMP*）
RFC 1157	*Simple Network Management Protocol*（*SNMP*）
RFC 1213	『*Management Information Base for Network Management of TCP/IP-based internets: MIB-II*』
RFC 1215	『*Convention for defining traps for use with the SNMP*』
RFC 1901	『*Introduction to Community-based SNMPv2*』
RFC 1905	『*Common Management Information Services and Protocol over TCP/IP (CMOT)*』
RFC 1906	『*Telnet X Display Location Option*』
RFC 1908	*Simple Network Management Protocol*（*SNMP*）
RFC 2104	『*HMAC: Keyed-Hashing for Message Authentication*』
RFC 2206	『*RSVP Management Information Base using SMIv2*』
RFC 2213	『*Integrated Services Management Information Base using SMIv2*』
RFC 2214	『*Integrated Service Management Information Base Guaranteed Service Extensions using SMIv2*』
RFC 2233	『The Interface Group MIB using SMIv2』
RFC 2271	『*An Architecture for Describing SNMP Management Frameworks*』
RFC 2570	『*Introduction to Version 3 of the Internet-standard Network Management Framework*』
RFC 2578	『*Structure of Management Information Version 2 (SMIv2)*』
RFC 2579	『*Textual Conventions for SMIv2*』
RFC 2580	『*Conformance Statements for SMIv2*』
RFC 2981	『Event MIB』

標準/RFC	Title
RFC 3413	『*SNMPv3 Applications*』
RFC 3415	『*View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)*』

MIB

MIB	MIB のリンク
• Cisco SNMPv2 • イーサネット ライク インターフェイス MIB • 『Event MIB』 • デルタ、ワイルドカード、集約の式 MIB サポート • インターフェイス グループ MIB（IF-MIB） • インターフェイス グループ MIB 機能拡張 • ユニバーサル ゲートウェイおよびアクセス サーバのための MIB 機能拡張	選択したプラットフォーム、Cisco IOS XE ソフトウェア リリース、およびフィーチャ セットの MIB の場所を検索しダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

シスコのテクニカル サポート

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

SNMP サポート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 *2* : *SNMP* サポートの設定に関する機能情報

機能名	リリース	機能情報
『Event MIB』	Cisco IOS XE Release 2.1	イベント MIB 機能は、Cisco ASR 1000 シリーズ ルータに実装されました。

機能名	リリース	機能情報
イベント MIB と式 MIB の CLI	Cisco IOS XE Release 3.1S	

機能名	リリース	機能情報
		イベント MIB と式 MIB 機能に、イベント MIB と式 MIB を設定するための CLI が導入されました。 この機能により、次のコマンドが導入または変更されました。 **action(event)**、**comparison**、**conditionalobject**、**delta(testthreshold)**、**deltainterval**、**description(event)**、**description(expression)**、**description(trigger)**、**discontinuityobject**、**enable(event)**、**enable(expression)**、**eventowner**、**enable(expression)**、**expression**、**falling(testthreshold)**、**frequency(eventtrigger)**、**object(expression)**、**object-id(actionnotification)**、**objectid(actionset)**、**objectid(eventtrigger)**、**objectlist(triggertest)**、**objectwildcard**、**rising(testthreshold)**、**sample(expression)**、**snmpmibeventobjectlist**、**snmpmibeventowner**、**snmpmibeventtrigger**、**snmpmibexpressiondelta**、**snmpmibexpressionowner**、**startup(testexistence)**、**startup(testboolean)**、**startup(testthreshold)**、**test(eventtrigger)**、**type(testexistence)**、**value(testboolean)**、**value(eventconfiguration)**、**valuetype**、

機能名	リリース	機能情報
		wildcard(eventandexpression)。
SNMP のためのインターフェイス インデックス表示	Cisco IOS XE Release 2.1	SNMP のためのインターフェイスインデックス表示の機能は、SNMP の上級ユーザがインターフェイスの登録に関する情報を管理対象エージェント上で直接表示できるようにするために、新しいコマンドとコマンドの変更を導入します。外部 NMS を使用しなくてもエージェントから MIB 情報を表示できます。 この機能は、インターフェイス MIB 内の 3 つのオブジェクト、*ifIndex*、*ifAlias*、および *ifName* を対象とします。これらのオブジェクトの完全な定義については、Cisco SNMPv2 MIB の Web サイト（ftp://ftp.cisco.com/pub/mibs/v2/）から入手できる IF-MIB.my ファイルを参照してください。
インターフェイス インデックスのパーシステンス	Cisco IOS XE Release 2.1	インターフェイス インデックス パーシステンスの拡張機能では、デバイスがリブートされても保持される一意の値でインターフェイスを識別できます。これらのインターフェイス ID 値は、SNMP を使用したネットワーク モニタリングとネットワーク管理に使用されます。
SNMP（簡易ネットワーク管理プロトコル）	Cisco IOS XE Release 2.1	

機能名	リリース	機能情報
SNMP 診断	Cisco IOS XE Release 3.1S	SNMP 診断機能で、最近ネットワーク管理システムによって要求されたオブジェクト ID を表示する Cisco IOS CLI コマンド、および SNMP デバッグ メッセージを表示する Cisco IOS CLI コマンドが追加されました。 次のコマンドが導入または変更されました。**showsnmpstatsoid** および **debugsnmpdetail**。
SNMP インフォーム要求	Cisco IOS XE Release 2.1	
SNMP マネージャ	Cisco IOS XE Release 2.1	SNMP マネージャ機能は、Cisco ASR 1000 シリーズ ルータに実装されました。
SNMP 通知ロギング	Cisco IOS XE Release 2.1	SNMP 通知ロギング機能は、コマンドラインで通知ログのサイズ変更、グローバルな ageout 値の設定、およびロギング サマリーの表示を行うための Cisco IOS CLI コマンドを追加します。
VPN のための SNMP サポート	Cisco IOS XE Release 2.1	VPN 機能の SNMP サポートでは、SNMP トラップおよびインフォームを VRF テーブルを使用して送受信できます。特に、この機能では、個々の VPN に固有の SNMP トラップおよびインフォームの送受信のために、Cisco IOS XE ソフトウェアでのサポートが追加されます。
SNMP バージョン 3	Cisco IOS XE Release 2.1	
SNMPv2C	Cisco IOS XE Release 2.1	

用語集

ifAlias：SNMP インターフェイス エイリアス。ifAlias は、IF-MIB のオブジェクトです。ifAlias は、ネットワーク マネージャによって指定されたインターフェイスのエイリアス名で、インターフェイスの不揮発性の説明を提供します。完全な定義については、IF-MIB.my ファイルを参照してください。

ifIndex：SNMP インターフェイス インデックス。ifIndex は、IF-MIB のオブジェクトです。ifIndex は、管理対象システム上のすべてのインターフェイス（サブインターフェイスも含む）に対して、インターフェイスが IF-MIB に登録されたときに割り当てられる一意の整数値です。完全な定義については、IF-MIB.my ファイルを参照してください。

OID：MIB オブジェクト ID。オブジェクト ID は、一連の整数値またはテキスト文字列で表されます。技術的には、数字形式はオブジェクト名で、テキスト形式はオブジェクトの説明です。実際には、両方ともオブジェクト ID（OID）と呼ばれます。たとえば、インターフェイス MIB のオブジェクト名は 1.3.6.1.2.1.2、オブジェクトの説明は「iso.internet.mgmt.mib-2.interfaces」であったりしますが、いずれも OID と呼ばれます。OID は、2 つの組み合わせ（iso.internet.2.1.2）で表すこともできます。

第 3 章

SNMP Support over VPNs：コンテキストベース アクセス コントロール

SNMP Support over VPNs：コンテキストベース アクセス コントロール機能は、SNMP コンテキスト サポートのインフラストラクチャを使用して、シスコ ソフトウェアでの複数のシンプル ネットワーク管理プロトコル（SNMP）のコンテキストのサポートに対応するインフラストラクチャ、および VPN 対応の MIB インフラストラクチャを提供します。

- 機能情報の確認，93 ページ
- SNMP Support over VPNs の制限事項：コンテキストベース アクセス コントロール，94 ページ
- SNMP Support over VPNs に関する情報：コンテキストベース アクセス コントロール，94 ページ
- SNMP Support over VPNs の設定方法--コンテキストベース アクセス コントロール，97 ページ
- SNMP Support over VPNs の設定例：コンテキスト ベース アクセス コントロール，102 ページ
- その他の参考資料，103 ページ
- SNMP Support over VPNs に関する機能情報：コンテキストベース アクセス コントロール，106 ページ

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、Bug Search Tool およびご使用の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。このモジュールで説明される機能に関する情報、および各機能がサポートされるリリースの一覧については、機能情報の表を参照してください。

プラットフォームのサポートおよびシスコ 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。

SNMP Support over VPNs の制限事項：コンテキストベース アクセス コントロール

- **nosnmp-servercontext** コマンドを使用して SNMP コンテキストを削除すると、そのコンテキストのすべての SNMP インスタンスが削除されます。
- すべての MIB が VPN に対応しているわけではありません。

SNMP Support over VPNs に関する情報：コンテキストベース アクセス コントロール

SNMP のバージョンとセキュリティ

Cisco ソフトウェアでは、次のバージョンの SNMP がサポートされています。

- SNMPv1：RFC 1157 で規定された、完全なインターネット標準のシンプル ネットワーク管理プロトコル（RFC 1157 は、RFC 1067 および RFC 1098 として公開されたそれより古いバージョンを置き換えるものです）。コミュニティ ストリングに基づいてセキュリティを実現します。
- SNMPv2c：SNMPv2 のためのコミュニティ ストリング ベースの管理フレームワーク。SNMPv2c（「c」は「community」を示す）は RFC 1901、RFC 1905、および RFC 1906 で規定されている実験的なインターネット プロトコルです。SNMPv2c は、SNMPv2p（SNMPv2 Classic）のプロトコル操作とデータ タイプが更新されたもので、SNMPv1 のコミュニティベースのセキュリティ モデルを使用します。

SNMP のバージョンに関する情報については、『*Cisco Network Management Configuration Guide*』の「Configuring SNMP Support」モジュールを参照してください。

SNMPv1 または SNMPv2 セキュリティ

Cisco IOS ソフトウェアでは、次のバージョンの SNMP がサポートされています。

- SNMPv1：RFC 1157 で規定された、完全なインターネット標準のシンプル ネットワーク管理プロトコル（RFC 1157 は、RFC 1067 および RFC 1098 として公開されたそれより古いバージョンを置き換えるものです）。コミュニティ ストリングに基づいてセキュリティを実現します。

- SNMPv2c：SNMPv2のためのコミュニティ ストリング ベースの管理フレームワーク。SNMPv2c（「c」は「community」を示す）は RFC 1901、RFC 1905、および RFC 1906 で規定されている実験的なインターネット プロトコルです。SNMPv2c は、SNMPv2p（SNMPv2 Classic）のプロトコル操作とデータ タイプが更新されたもので、SNMPv1 のコミュニティベースのセキュリティ モデルを使用します。

SNMPv1 と SNMPv2 は、SNMPv3 ほど安全ではありません。SNMP バージョン 1 と 2 では、プレーンテキスト コミュニティを使用し、SNMP バージョン 3 で実行される認証やセキュリティ チェックを実行しません。SNMP バージョン 1 または SNMP バージョン 2 を使用するときに SNMP Support over VPNs：コンテキストベース アクセス コントロール機能を設定するには、コミュニティ名を VPN に関連付ける必要があります。関連付けると、SNMP は、特定のコミュニティ ストリングの着信要求を、設定されている VRF から受信した場合だけ処理します。着信パケットに含まれているコミュニティ ストリングに VRF が関連付けられていない場合は、VRF 以外のインターフェイス経由で着信した場合だけパケットを処理します。このプロセスによって、VPN 外のユーザがクリア テキスト コミュニティ ストリングをスヌーピングして VPN データを照会するのを防ぐことができます。送信元アドレス確認のこれらの方法は、SNMPv3 を使用するほど安全ではありません。

SNMP Notification Support over VPNs

SNMP Notification Support over VPNs 機能では、VPN ルーティングおよび転送（VRF）インスタンス テーブルを使用して SNMP 通知（トラップおよびインフォーム）の送信および受信を行えます。特に、この機能では、個々の VPN に固有の SNMP 通知（トラップおよびインフォーム）の送受信のために、Cisco ソフトウェアでのサポートが追加されます。

SNMP は、アプリケーション層プロトコルであり、SNMP マネージャと SNMP エージェントとの通信に使用されるメッセージ フォーマットを提供します。

VPN は、プライベート ネットワークと同じ使用ガイドラインを持つ共有システム上での接続性の高い転送を提供するネットワークです。VPN は IP でのインターネット、フレーム リレー、または ATM ネットワーク上で構築できます。

VRF では VPN ごとにルーティング データを保存します。これはネットワーク アクセス サーバ（NAS）に接続された顧客のサイトの VPN メンバーシップを定義します。VRF は、IP ルーティング テーブル、派生する Cisco Express Forwarding（以前の CEF）テーブル、ガイドラインおよびルーティング テーブルに含まれる情報を制御するルーティング プロトコル パラメータで構成されます。

VPN のための SNMP サポート：コンテキストベース アクセス コントロール機能は、ユーザが SNMP エージェントとマネージャを特定の VRF と関連付けることを可能にするコンフィギュレーション コマンドを提供します。関連付けられた VRF は、エージェントとマネージャの間の SNMP 通知（トラップおよびインフォーム）と応答の送信に使用されます。VRF が指定されなかった場合は、その VPN のデフォルトのルーティング テーブルが使用されます。

VPN 対応 SNMP

VPN の SNMP サポート：コンテキストベース アクセス コントロール機能により、SNMP Notification Support for VPNs 機能が拡張され、異なる VPN からの着信パケットを SNMP で区別できるようになります。

VPN の SNMP サポート時：コンテキストベース アクセス コントロール機能が設定され、SNMP は、設定された VRF 要求を受け付け、同じ VRF に応答を返します。トラップ ホストは、特定の VRF に関連付けることもできます。これにより、設定された VRF を使用してトラップが送信されます。関連付けしない場合は、デフォルトのルーティング テーブルが使用されます。特定の VRF にリモート ユーザを関連付けることもできます。また、SNMP が要求を受け入れる必要がある VRF を設定できます。指定されていない VRF から送信される要求はドロップされます。

IP アクセス リストを設定し、SNMP コミュニティ ストリングに関連付けることができます。この機能を使用すると、VRF インスタンスと SNMP コミュニティ ストリング間のアソシエーションを設定できます。VRF インスタンスが SNMP コミュニティ ストリングに関連付けられている場合、SNMP はその要求が設定されている VRF から受信した場合だけ、特定のコミュニティ ストリングの着信要求を処理します。着信パケットに含まれているコミュニティ ストリングに関連付けられている VRF がない場合、コミュニティ ストリングは VRF 以外のインターフェイス経由で着信した場合だけ処理されます。

VRF のミスマッチによりドロップされた SNMP パケットの認証トラップをイネーブルまたはディセーブルにすることもできます。デフォルトでは、SNMP 認証トラップをイネーブルすると、VRF 認証トラップもイネーブルになります。

VPN ルート識別子

ルート識別子（RD）はルーティング テーブルとフォワーディング テーブルを作成し、VPN のデフォルトのルート識別子を指定します。RD は顧客の IPv4 プレフィックスの先頭に追加され、その IPv4 プレフィックスはグローバルに一意の VPN-IPv4 プレフィックスになります。

RD は、自律システム番号と任意の番号で構成される自律システム番号（ASN）相対 RD、または IP アドレスと任意の番号で構成される IP アドレス相対 RD のいずれかです。

RD は、次のいずれかの形式で入力できます。

- 16 ビット ASN：101:3 などの 16 ビット数値
- 32 ビット IP アドレス：192.168.122.15:1 などの 32 ビット数値

SNMP コンテキスト

SNMP コンテキストによって、MIB データにアクセスする安全な方法が VPN ユーザに提供されます。VPN がコンテキストに関連付けられると、VPN 固有の MIB データがそのコンテキストに存在します。VPN をコンテキストに関連付けると、サービス プロバイダーが、複数 VPN でネットワークを管理できます。コンテキストを作成して VPN に関連付けることにより、プロバイダー

は、ある VPN のユーザが同じネットワーキング デバイス上で他の VPN のユーザに関する情報にアクセスするのを防ぐことができます。

VPN 対応 SNMP で、SNMP セキュリティ名と VPN ID のマッピングの VPN 環境で実行する SNMP マネージャとエージェント エンティティ間の契約が必要です。このマッピングは、SNMP-VACM-MIB の設定を通じて、異なる VPN の SNMP データ用に複数のコンテキストを使用して作成されます。セキュリティ名を持つ VPN 上のユーザがその VPN のユーザに関連付けられたコンテキストのユーザのアクセス タイプに関連づけられた制限付きオブジェクト領域にアクセスできるように、SNMP-VACM-MIB がビューとともに設定されます。

応答メッセージが、VPN コンテキスト内のオブジェクト値を使用して返信される前に、SNMP 要求メッセージは、セキュリティとアクセス コントロールの次のような 3 つのフェーズを経ます。

- 最初のフェーズでは、ユーザ名が認証されます。このフェーズでは、ユーザが SNMP アクセスに対して認証および許可されるようにします。
- 第 2 フェーズでは、ユーザは、設定されている SNMP コンテキストの考慮時にグループ オブジェクトに対して要求された SNMP アクセス用に承認されます。このフェーズは、アクセス制御フェーズと呼ばれます。
- 第 3 フェーズでは、テーブル エントリの特定のインスタンスへのアクセスが行われます。この 3 つ目のフェーズでは、完全な取得は、SNMP コンテキスト名を使用して行えます。

SNMP Support over VPNs の設定方法--コンテキストベース アクセス コントロール

SNMP コンテキストの設定および SNMP コンテキストと VPN の関連付け

SNMP コンテキストを設定し、VPN と SNMP コンテキストを関連付けるには、この作業を実行します。

（注）
- 次の MIB コンテキストだけを認識します。これらの MIB のすべてのテーブルをポーリングできます。
  - CISCO-IPSEC-FLOW-MONITOR-MIB
  - CISCO-IPSEC-MIB
  - CISCO-PING-MIB
  - IP-FORWARD-MIB
  - MPLS-LDP-MIB
- IP-FORWARD-MIB の 2 個の SNMP 変数 1.3.6.1.2.1.4.24.3（ipCidrRouteNumber：スカラ）と 1.3.6.1.2.1.4.24.4.1（ipCidrRouteEntry：テーブル）だけをポーリングできます。

手順の概要

1. イネーブル化
2. **configureterminal**
3. **snmp-servercontext***context-name*
4. **ipvrf***vrf-name*
5. **rd***route-distinguisher*
6. **context***context-name*
7. **route-target** {**import** | **export** | **both**} *route-target-ext-community*
8. **end**
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	イネーブル化 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	**snmp-servercontext***context-name* 例： `Device(config)# snmp-server context context1`	SNMP コンテキストを作成し、その名前を指定します。
ステップ 4	**ipvrf***vrf-name* 例： `Device(config)# ip vrf vrf1`	VRF ルーティング テーブルを設定し、VRF 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	**rd***route-distinguisher* 例： `Device(config-vrf)# rd 100:120`	VPN ルート識別子を作成します。
ステップ 6	**context***context-name* 例： `Device(config-vrf)# context context1`	SNMP コンテキストを特定の VRF に関連付けます。 （注） リリースに応じて、**context** コマンドは **snmpcontext** コマンドに置き換えられます。詳細については、『*Cisco IOS Network Management Command Reference*』を参照してください。
ステップ 7	**route-target** {**import** | **export** | **both**} *route-target-ext-community* 例： `Device(config-vrf)# route-target export 100:1000`	（任意）VRF 用の route-target 拡張コミュニティを作成します。
ステップ 8	**end** 例： `Device(config-vrf)# end`	インターフェイス モードを終了し、グローバル コンフィギュレーション モードを開始します。
ステップ 9	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了します。

SNMP サポートの設定および SNMP コンテキストの関連付け

SNMP サポートの設定と SNMP コンテキストへの関連付けを行うには、この作業を実行します。

手順の概要

1. イネーブル化
2. **configureterminal**
3. **snmp-serveruser***usernamegroup-name* [**remote***host* [**udp-port***port*] [**vrf***vrf-name*]] {**v1** | **v2c** | **v3** [**encrypted**] [**auth** {**md5** | **sha**} *auth-password*]} [**access** [**ipv6***nacl*] [**priv** {**des** | **3des** | **aes** {**128** | **192** | **256**}} *privpassword*] {*acl-number* | *acl-name*}]
4. **snmp-servergroup***group-name* {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}} [**context***context-name*] [**read***read-view*] [**write***write-view*] [**notify***notify-view*] [**access** [**ipv6***named-access-list*] [*acl-number*| *acl-name*]]
5. **snmp-serverview***view-nameoid-tree* {**included** | **excluded**}
6. **snmp-serverenabletraps** [*notification-type*] [**vrrp**]
7. **snmp-servercommunity***string* [**view***view-name*] [**ro** | **rw**] [**ipv6***nacl*] [*access-list-number* | *extended-access-list-number* | *access-list-name*]
8. **snmp-serverhost** {*hostname* | *ip-address*} [**vrf***vrf-name*] [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port***port*] [*notification-type*]
9. **snmpmibcommunity-map***community-name* [**context***context-name*] [**engineid***engine-id*] [**security-name***security-name*][**target-list***upn-list-name*]
10. **snmpmibtargetlist***vpn-list-name* {**vrf***vrf-name* | **host***ip-address*}
11. **nosnmp-servertrapauthenticationvrf**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	イネーブル化 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**snmp-serveruser***usernamegroup-name* [**remote***host* [**udp-port***port*] [**vrf***vrf-name*]] {**v1**	**v2c**	**v3** [**encrypted**] [**auth** {**md5**	**sha**} *auth-password*]} [**access** [**ipv6***nacl*] [**priv** {**des**	**3des**	**aes** {**128**	**192**	**256**}} *privpassword*] {*acl-number*	*acl-name*}]	SNMP グループに新しいユーザを設定します。

	コマンドまたはアクション	目的
	例： `Device(config)# snmp-server user customer1 group1 v1`	
ステップ 4	**snmp-servergroup***group-name* {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}} [**context***context-name*] [**read***read-view*] [**write***write-view*] [**notify***notify-view*] [**access** [**ipv6***named-access-list*] [*acl-number*| *acl-name*]] 例： `Device(config)# snmp-server group group1 v1 context context1 read view1 write view1 notify view1`	新規 SNMP グループ、または SNMP ユーザを SNMP ビューにマッピングするテーブルを設定します。 • **context***context-name* キーワードと引数のペアを使用して、指定した SNMP グループを設定されている SNMP コンテキストに関連付けます。
ステップ 5	**snmp-serverview***view-name oid-tree* {**included** | **excluded**} 例： `Device(config)# snmp-server view view1 ipForward included`	ビュー エントリを作成または更新します。
ステップ 6	**snmp-serverenabletraps** [*notification-type*] [**vrrp**] 例： `Device(config)# snmp-server enable traps`	システムで使用できるすべての SNMP 通知（トラップまたは応答要求）をイネーブルにします。
ステップ 7	**snmp-servercommunity***string* [**view***view-name*] [**ro** | **rw**] [**ipv6***nacl*] [*access-list-number* | *extended-access-list-number* | *access-list-name*] 例： `Device(config)# snmp-server community public view view1 rw`	SNMP へのアクセスを許可するコミュニティ アクセス ストリングを設定します。
ステップ 8	**snmp-serverhost** {*hostname* | *ip-address*} [**vrf***vrf-name*] [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port***port*] [*notification-type*] 例： `Device(config)# snmp-server host 10.0.0.1 vrf vrf1 public udp-port 7002`	SNMP 通知動作の指定

	コマンドまたはアクション	目的
ステップ 9	**snmpmibcommunity-map***community-name* [**context***context-name*] [**engineid***engine-id*] [**security-name***security-name*][**target-list***upn-list-name*] 例： `Device(config)# snmp mib community-map community1 context context1 target-list commAVpn`	SNMP コミュニティを SNMP コンテキスト、エンジン ID、またはセキュリティ名にマッピングします。
ステップ 10	**snmpmibtargetlist***vpn-list-name* {**vrf***vrf-name* | **host***ip-address*} 例： `Device(config)# snmp mib target list commAVpn vrf vrf1`	SNMP コミュニティに関連付けるターゲット VRF とホストのリストを作成します。
ステップ 11	**nosnmp-servertrapauthenticationvrf** 例： `Device(config)# no snmp-server trap authentication vrf`	（任意）VRF インターフェイスで受信したパケットについて生成されるすべての SNMP 認証通知（トラップまたは応答要求）をディセーブルにします。 • このコマンドを使用して、関連付けられているコミュニティが正しくない VRF インターフェイス上のパケットに対する認証トラップだけをディセーブルにします。

SNMP Support over VPNs の設定例：コンテキスト ベース アクセス コントロール

例：コンテキストベース アクセス コントロールの設定

次の設定例では、SNMP Support over VPNs（SNMPv1 または SNMPv2 のコンテキストベース アクセス コントロール機能）の設定方法を示します。

（注） リリースに応じて、**context** コマンドは **snmpcontext** コマンドに置き換えられます。詳細については、『*Cisco IOS Network Management Command Reference*』を参照してください。

```
snmp-server context A
```

```
snmp-server context B
ip vrf Customer_A
 rd 100:110
 context A
 route-target export 100:1000
 route-target import 100:1000
!
ip vrf Customer_B
 rd 100:120
 context B
 route-target export 100:2000
 route-target import 100:2000
!
interface Ethernet3/1
 description Belongs to VPN A
 ip vrf forwarding CustomerA
 ip address 192.168.2.1 255.255.255.0

interface Ethernet3/2
 description Belongs to VPN B
 ip vrf forwarding CustomerB
 ip address 192.168.2.2 255.255.255.0
snmp-server user commA grp1A v1
snmp-server user commA grp2A v2c
snmp-server user commB grp1B v1
snmp-server user commB grp2B v2c
snmp-server group grp1A v1 context A read viewA write viewA notify viewA
snmp-server group grp1B v1 context B read viewB write viewB notify viewB
snmp-server view viewA ipForward included
snmp-server view viewA ciscoPingMIB included
snmp-server view viewB ipForward included
snmp-server view viewB ciscoPingMIB included
snmp-server enable traps
snmp-server host 192.168.2.3 vrf CustomerA commA udp-port 7002
snmp-server host 192.168.2.4 vrf CustomerB commB udp-port 7002
snmp mib community-map commA context A target-list commAvpn
! Configures source address validation
snmp mib community-map commB context B target-list commBvpn
! Configures source address validation
snmp mib target list commAvpn vrf CustomerA
! Configures a list of VRFs or from which community commA is valid
snmp mib target list commBvpn vrf CustomerB
! Configures a list of VRFs or from which community commB is valid
```

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
Cisco IOS コマンド	『Cisco IOS Master Commands List, All Releases』
『Cisco IOS SNMP Support Command Reference』	『Cisco IOS SNMP Support Command Reference』

標準および RFC

標準/RFC	Title
CBC-DES（DES-56）標準	『*Symmetric Encryption Protocol*』

標準/RFC	Title
規格 58	『*Structure of Management Information Version 2 (SMIv2)*』>
RFC 1067	『*A Simple Network Management Protocol*』
RFC 1091	『*Telnet terminal-type option*』
RFC 1098	*Simple Network Management Protocol*（*SNMP*）
RFC 1157	*Simple Network Management Protocol*（*SNMP*）
RFC 1213	『*Management Information Base for Network Management of TCP/IP-based internets: MIB-II*』
RFC 1215	『*Convention for defining traps for use with the SNMP*』
RFC 1901	『*Introduction to Community-based SNMPv2*』
RFC 1905	『*Common Management Information Services and Protocol over TCP/IP (CMOT)*』
RFC 1906	『*Telnet X Display Location Option*』
RFC 1908	*Simple Network Management Protocol*（*SNMP*）
RFC 2104	『*HMAC: Keyed-Hashing for Message Authentication*』
RFC 2206	『*RSVP Management Information Base using SMIv2*』
RFC 2213	『*Integrated Services Management Information Base using SMIv2*』
RFC 2214	『*Integrated Service Management Information Base Guaranteed Service Extensions using SMIv2*』
RFC 2233	『The Interface Group MIB using SMIv2』
RFC 2271	『*An Architecture for Describing SNMP Management Frameworks*』
RFC 2570	『*Introduction to Version 3 of the Internet-standard Network Management Framework*』
RFC 2578	『*Structure of Management Information Version 2 (SMIv2)*』
RFC 2579	『*Textual Conventions for SMIv2*』
RFC 2580	『*Conformance Statements for SMIv2*』
RFC 2981	『Event MIB』

標準/RFC	Title
RFC 3413	『*SNMPv3Applications*』
RFC 3415	『*View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)*』

MIB

MIB	MIB のリンク
• Cisco SNMPv2 • イーサネット ライク インターフェイス MIB • 『Event MIB』 • デルタ、ワイルドカード、集約の式 MIB サポート • インターフェイス グループ MIB（IF-MIB） • インターフェイス グループ MIB 機能拡張 • ユニバーサル ゲートウェイおよびアクセス サーバのための MIB 機能拡張	選択したプラットフォーム、Cisco IOS XE ソフトウェア リリース、およびフィーチャ セットの MIB の場所を検索しダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

シスコのテクニカル サポート

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

SNMP Support over VPNs に関する機能情報：コンテキストベース アクセス コントロール

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*：*SNMP Support over VPNs* に関する機能情報：コンテキストベース アクセス コントロール

機能名	リリース	機能情報
SNMP Support over VPNs：コンテキストベース アクセス コントロール		SNMP Support over VPNs：コンテキスト ベース アクセス コントロール機能は、シスコ ソフトウェアの複数の SNMP コンテキスト サポートのインフラストラクチャと複数の SNMP コンテキスト サポート インフラストラクチャを使用する VPN 対応 MIB インフラストラクチャを提供します。

第 4 章

SNMP バージョン 3 のための AES および 3-DES 暗号化サポート

SNMP バージョン 3 機能の AES および 3 DES 暗号化サポートは、Simple Network Management Protocol（SNMP）バージョン 3 の暗号化機能を強化します。

SNMP バージョン 3 機能の AES および 3 DES 暗号化サポートは、RFC 3826 に準拠した Advanced Encryption Standard（AES）128 ビット暗号化を追加します。

- 機能情報の確認，107 ページ
- SNMP バージョン 3 の AES および 3 DES 暗号化サポートの前提条件，108 ページ
- SNMP バージョン 3 のための AES および 3 DES 暗号化サポートに関する情報，108 ページ
- SNMP バージョン 3 のための AES および 3 DES 暗号化サポートの設定方法，110 ページ
- その他の参考資料，112 ページ
- SNMP バージョン 3 の AES および 3 DES 暗号化サポートの機能情報，113 ページ

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、Bug Search Tool およびご使用の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。このモジュールで説明される機能に関する情報、および各機能がサポートされるリリースの一覧については、機能情報の表を参照してください。

プラットフォームのサポートおよびシスコ 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。

SNMP バージョン 3 の AES および 3 DES 暗号化サポートの前提条件

- ネットワーク管理ステーション（NMS）は、この機能を使用できるように Simple Network Management Protocol（SNMP）バージョン 3 をサポートしている必要があります。
- この機能は、暗号化アルゴリズムをサポートする Cisco ソフトウェア イメージでだけ使用可能です。
- 使用されるセキュリティ モデルとアーキテクチャ内の他のサブシステムとのセキュリティ モデルの相互作用について把握するには、SNMP アーキテクチャとアーキテクチャの用語を理解することが重要です。

SNMP バージョン 3 のための AES および 3 DES 暗号化サポートに関する情報

Cipher Block Chaining/Data Encryption Standard（CBC-DES）は、SNMP バージョン 3 機能の AES および 3 DES 暗号化サポートのプライバシー プロトコルです。この機能の導入前は、DES だけがサポートされていました（RFC 3414 に準拠）。この機能は、AES-128（RFC 3826 による）および AES-192、AES-256、および 3-DES（CISCO-SNMP-USM-OIDS-MIB による）のサポートを追加します。RFC 3826 機能拡張が SNMP-USM-AES-MIB に含められました。さらに、トリプル データ暗号化規格（3-DES）、および AES 192 ビットおよび 256 ビット 暗号化をサポートするための Cisco 固有の機能拡張が、CISCO-SNMP-USM-MIB に追加されました。追加情報は、「Extension to the User-Based Security Model (USM) to Support Triple-DES EDE in "Outside" CBC Mode」という名前のインターネットドラフトにあります。

暗号キーのサイズは次のとおりです。

- AES 暗号化は、暗号キー サイズ 128 ビット、192 ビット、または 256 ビットでの暗号フィードバック（CFB）モードを使用します。
- 3-DES 暗号化は、暗号化に 168 ビットのキー サイズを使用します。

「The AES Cipher Algorithm in the Simple Network Management Protocol (SNMP) User-based Security Model (USM)」ドラフトでは、128 ビットのキー サイズでの AES の使用について説明されています。ただし、この機能拡張で、USM を使用するためのその他のオプションも実装されます。AES の 192 ビットまたは 256 ビット サイズのキーまたは 3-DES の 168 ビット サイズのキーについて、ローカライズされたキーを生成する規格はありません。長いキーには、使用できる認証プロトコルはありません。

SNMP バージョン 3 USM のサポートは、RFC 3414 に準拠しています。RFC 3414 は、DES を SNMP バージョン 3 authPriv モードのメッセージ暗号化に必要な唯一の方法として規定します。

SNMP バージョン 3 機能の AES および 3 DES 暗号化サポートは、CLI および MIB によるプライバシー プロトコルの選択をサポートしています。新しい標準 MIB である SNMP-USM-AES-MIB は、Advanced Encryption Standard（AES）での 128 ビット キーのサポートを提供します。192 ビットまたは 256 ビット キーでの AES および 3-DES の拡張オプションは、SNMP-USM-MIB の機能拡張として、シスコ固有の MIB である CISCO-SNMP-USM-EXT-MIB でサポートされます。

AES および 3-DES 暗号化サポートの概要

Simple Network Management Protocol（SNMP）の各エンティティに 1 つの SNMP エンジンが含まれます。SNMP エンジンは、メッセージの送信と受信、メッセージの認証と暗号化/復号化、および管理対象オブジェクトへのアクセス制御を行う機能を実装します。これらの機能は、SNMP エンティティを作成するために SNMP エンジンで設定されている 1 つまたは複数のアプリケーションでサービスとして提供されます。RFC 3411 では、次のコンポーネントで構成されるように SNMP エンジンを説明しています。

- ディスパッチャ
- メッセージ処理サブシステム
- セキュリティ サブシステム
- アクセス コントロール サブシステム

Cipher Block Chaining/Data Encryption Standard（CBC-DES）は、SNMP バージョン 3 機能の AES および 3 DES 暗号化サポートのプライバシー プロトコルです。この機能の導入前は、DES だけがサポートされていました（RFC 3414 に準拠）。この機能は、AES-128（RFC 3826 による）および AES-192、AES-256、および 3-DES（CISCO-SNMP-USM-OIDS-MIB による）のサポートを追加します。RFC 3826 機能拡張が SNMP-USM-AES-MIB に含められました。さらに、トリプル データ暗号化規格（3-DES）、および AES 192 ビットおよび 256 ビット 暗号化をサポートするための Cisco 固有の機能拡張が、CISCO-SNMP-USM-MIB に追加されました。追加情報は、「Extension to the User-Based Security Model (USM) to Support Triple-DES EDE in "Outside" CBC Mode」という名前のインターネットドラフトにあります。

暗号キーのサイズは次のとおりです。

- AES 暗号化は、暗号キー サイズ 128 ビット、192 ビット、または 256 ビットでの暗号フィードバック（CFB）モードを使用します。
- 3-DES 暗号化は、暗号化に 168 ビットのキー サイズを使用します。

「The AES Cipher Algorithm in the Simple Network Management Protocol (SNMP) User-based Security Model (USM)」ドラフトでは、128 ビットのキー サイズでの AES の使用について説明されています。ただし、この機能拡張で、USM を使用するためのその他のオプションも実装されます。AES の 192 ビットまたは 256 ビット サイズのキーまたは 3-DES の 168 ビット サイズのキーについて、ローカライズされたキーを生成する規格はありません。長いキーには、使用できる認証プロトコルはありません。

SNMP バージョン 3 USM のサポートは、RFC 3414 に準拠しています。RFC 3414 は、DES を SNMP バージョン 3 authPriv モードのメッセージ暗号化に必要な唯一の方法として規定します。

SNMP バージョン 3 機能の AES および 3 DES 暗号化サポートは、CLI および MIB によるプライバシー プロトコルの選択をサポートしています。新しい標準 MIB である SNMP-USM-AES-MIB は、Advanced Encryption Standard（AES）での 128 ビット キーのサポートを提供します。192 ビットまたは 256 ビット キーでの AES および 3-DES の拡張オプションは、SNMP-USM-MIB の機能拡張として、シスコ固有の MIB である CISCO-SNMP-USM-EXT-MIB でサポートされます。

暗号キーのサポート

MIB サポート

SNMP バージョン 3 のための AES および 3 DES 暗号化サポートの設定方法

SNMP グループへの新しいユーザの追加

手順の概要

1. イネーブル化
2. **configureterminal**
3. **snmp-serveruser***usernamegroup-name* [**remote***host* [**udp-port***port*]] {**v1** | **v2c** | **v3** [**encrypted**] [**auth** {**md5** | **sha**} *auth-password*]} [**priv** {**des** | **3des** | **aes{128|192|256}**} *privpassword*] [**access** [**ipv6***nacl*] {*acl-number* | *acl-name*}]
4. **exit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	イネーブル化 例： `Device> enable`	特権 EXEC モードを開始します。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	**snmp-serveruser***usernamegroup-name* [**remote***host* [**udp-port***port*]] {**v1** | **v2c** | **v3** [**encrypted**] [**auth** {**md5** | **sha**} *auth-password*]} [**priv** {**des** | **3des** | **aes{128|192|256}**} *privpassword*] [**access** [**ipv6***nacl*] {*acl-number* | *acl-name*}] 例： `Device(config)# snmp-server user new-user new-group v3 auth md5 secureone priv aes 128 privatetwo access 2`	SNMP ユーザを追加し、そのユーザが属するグループを指定し、使用する権限付与アルゴリズム（MD5 または SHA）を指定し、使用するプライバシーアルゴリズム（DES、3-DES、AES、AES-192、または AES-256）を指定し、そのプライバシープロトコルに関連付けるパスワードを指定します。
ステップ 4	**exit** 例： `Device(config)# exit`	グローバルコンフィギュレーションモードを終了し、特権 EXEC モードに戻ります。

SNMP ユーザの設定の確認

Simple Network Management Protocol（SNMP）ユーザの設定されている特性に関する情報を表示するには、特権 EXEC モードで **showsnmpuser** コマンドを使用します。

（注） **showsnmpuser** コマンドは、デバイス上で設定されているすべてのユーザを表示します。ただし、他の SNMP 設定と違って、**snmp-serveruser** コマンドは「show running」出力には現れません。

手順の概要

1. イネーブル化
2. **showsnmpuser** [*username*]

手順の詳細

ステップ 1 イネーブル化

例：

```
Device> enable
```

特権 EXEC モードを開始します。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 **showsnmpuser** [*username*]

例：

```
Device# show snmp user abcd

User name: abcd
Engine ID: 000000090200000000C025808
storage-type: nonvolatile       active access-list: 10
Rowstatus: active
Authentication Protocol: MD5
Privacy protocol: 3DES
Group name: VacmGroupName
Group name: VacmGroupName
```

上の例では、ユーザ名に abcd、エンジン ID 文字列に 000000090200000000C025808、ストレージ タイプに nonvolatile が指定されています。

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
Cisco IOS コマンド	
コマンド	

標準

規格	Title

MIB

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

RFC

RFC	Title

シスコのテクニカル サポート

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

SNMP バージョン 3 の AES および 3 DES 暗号化サポートの機能情報

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 *4* : ***SNMP*** バージョン *3* の ***AES*** および ***3 DES*** 暗号化サポートの機能情報

機能名	リリース	機能情報
SNMP バージョン 3 のための AES および 3-DES 暗号化サポート		SNMP バージョン 3 機能の AES および 3 DES 暗号化サポートは、Simple Network Management Protocol（SNMP）バージョン 3 の暗号化機能を強化します。データ暗号規格（DES）サポートは、Cisco IOS Release 12.0 で導入され、Cisco IOS Release 12.1 で拡張されました。SNMP 3 User-based Security Model（USM）のサポートは、SNMP バージョン 3 authPriv モードのメッセージ暗号化に必要な唯一の方法として DES を定義する RFC 3414 に準拠します。

第 5 章

VLAN サブインターフェイスに対する簡易ネットワーク管理プロトコル（SNMP）サポート

このフィーチャ モジュールでは、VLAN サブインターフェイス機能に対する SNMP サポートについて説明します。新機能の利点、サポートされるプラットフォーム、サポートされる規格、VLAN サブインターフェイス機能に対する SNMP サポートを設定するために必要なコマンドについて説明します。

VLAN サブインターフェイスに対する SNMP サポート機能は、ファスト イーサネット サブインターフェイスに mib-2 インターフェイス スパース テーブル サポートを提供します。この機能拡張は、フレーム リレー サブインターフェイスでサポートされる機能に似ています。

- 機能情報の確認，115 ページ
- VLAN サブインターフェイスに対する SNMP サポートについて，116 ページ
- VLAN サブインターフェイスに対する SNMP サポートの仕組み，116 ページ
- VLAN サブインターフェイスでの SNMP サポートの設定例，118 ページ
- その他の参考資料，119 ページ
- VLAN サブインターフェイスでの SNMP サポートの機能情報，120 ページ

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、Bug Search Tool およびご使用の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。このモジュールで説明される機能に関する情報、および各機能がサポートされるリリースの一覧については、機能情報の表を参照してください。

プラットフォームのサポートおよびシスコ 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。

VLAN サブインターフェイスに対する SNMP サポートについて

利点

ファスト イーサネット サブインターフェイス上のインターフェイス テーブルにスパース テーブルをサポートすることで、フレーム リレー サブインターフェイスに慣れていたユーザに同じ機能が提供されます。

サポートされるプラットフォーム

- Cisco 2600 シリーズ
- Cisco 3600 シリーズ
- Cisco 4000-m シリーズ
- Cisco 7200 シリーズ
- Cisco 7500 シリーズ

VLAN サブインターフェイスに対する SNMP サポートの仕組み

VLAN サブインターフェイスでの SNMP エージェントのイネーブル化

VLAN サブインターフェイスで SNMP エージェントをイネーブルにするには、次の作業を実行します。

手順の概要

1. イネーブル化
2. **configureterminal**
3. **snmpcommunity***string*
4. **interface***typeslot/port*
5. **encapsulationisl***vlan-identifier*
6. **ipaddress***ip-addressmask*
7. **end**
8. **showvlans**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	イネーブル化 例： `Router> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Router# 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	**snmpcommunity***string* 例： `Router(config)# snmp community public`	リモート アクセスの SNMP エージェントをイネーブルにします。
ステップ 4	**interface***typeslot/port* 例： `Router(config)# interface FastEthernet 0/1.1`	設定するファスト イーサネット インターフェイスを選択します。
ステップ 5	**encapsulationisl***vlan-identifier* 例： `Router(config-if)#  encapsulation isl 10`	Inter-Switch リンク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 6	**ipaddress***ip-addressmask* 例： `Router(config)# ip address 192.168.10.1 255.255.255.0`	インターフェイスに対するプライマリ IP アドレスまたはセカンダリ IP アドレスを設定します。
ステップ 7	**end** 例： `Router(config-if)# end`	特権 EXEC モードに戻ります。
ステップ 8	**showvlans** 例： `Router# show vlans`	VLAN サブインターフェイスを表示します。

VLAN サブインターフェイスでの SNMP サポートの設定例

例：VLAN サブインターフェイスでの SNMP エージェントのイネーブル化

次の設定例は、VLAN サブインターフェイス付属のルータで SNMP エージェントをイネーブルにして、SNMP アプリケーションをリモートで監視する方法を示しています。

```
snmp community public
!
interface FastEthernet4/0.100
 encapsulation isl 100
 ip address 192.168.10.21 255.255.255.0
!
interface FastEthernet4/0.200
 encapsulation isl 200
 ip address 172.21.200.11 255.255.255.0
!
interface FastEthernet4/1.1
 encapsulation isl 10
 ip address 171.69.2.111 255.255.255.0
```

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
Cisco IOS コマンド	『Cisco IOS Master Commands List, All Releases』
SNMP コマンド	『*Cisco IOS Network Management Command Reference*』

標準

規格	Title
なし	--

MIB

MIB	MIB のリンク
• なし	選択したプラットフォーム、Cisco ソフトウェア リリース、およびフィーチャ セットの MIB を検索してダウンロードする場合は、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

RFC

RFC	Title
RFC 1573	*Evolution of the Interfaces Group of MIB-II*

シスコのテクニカル サポート

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

VLAN サブインターフェイスでの SNMP サポートの機能情報

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 *5* : *VLAN* サブインターフェイスでの *SNMP* サポートの機能情報

機能名	リリース	機能情報
VLAN サブインターフェイスに対する簡易ネットワーク管理プロトコル（SNMP）サポート	12.2	VLAN サブインターフェイスに対する SNMP サポート機能は、ファスト イーサネット サブインターフェイスに mib-2 インターフェイス スパース テーブル サポートを提供します。この機能拡張は、フレーム リレー サブインターフェイスでサポートされる機能に似ています。

第 6 章

メモリ プール：SNMP 通知のサポート

この機能は Cisco 拡張メモリ プール MIB（CISCO-ENHANCED-MEMPOOL-MIB）の SNMP 通知をイネーブルにする CLI コマンドを追加します。

- 機能情報の確認，121 ページ
- メモリ プールの前提条件：SNMP 通知のサポート，122 ページ
- メモリ プールの制限事項：SNMP 通知のサポート，122 ページ
- メモリ プールに関する情報：SNMP 通知のサポート，122 ページ
- メモリ プールをイネーブルにする方法：SNMP 通知のサポート，123 ページ
- メモリ プールの設定例：SNMP 通知のサポート，124 ページ
- その他の参考資料，124 ページ
- メモリ プールの機能情報：SNMP 通知のサポート，127 ページ

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、Bug Search Tool およびご使用の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。このモジュールで説明される機能に関する情報、および各機能がサポートされるリリースの一覧については、機能情報の表を参照してください。

プラットフォームのサポートおよびシスコ 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。

メモリ プールの前提条件：SNMP 通知のサポート

CISCO-ENHANCED-MEMPOOL-MIB をコンパイルする前に、リストされている順序で次の MIB をコンパイルする必要があります。

1. SNMPv2-SM（SNMP 設定 MIB）
2. SNMPv2-TC（SNMP 設定 MIB）
3. SNMPv2-CONF（SNMP 設定 MIB）
4. SNMP-FRAMEWORK-MIB（SNMP 設定 MIB）
5. CISCO-SMI（SNMP 設定 MIB）
6. ENTITY-MIB（コア MIB）
7. CISCO-ENHANCED-MEMPOOL-MIB（インフラストラクチャ MIB）

シスコ デバイスで使用されるすべての MIB は、http://www.cisco.com/go/mibs で入手できます。

メモリ プールの制限事項：SNMP 通知のサポート

MIB へのアクセスは読み取り専用レベルに制限されます。

メモリ プールに関する情報：SNMP 通知のサポート

CISCO-ENHANCED-MEMPOOL-MIB モジュールでは、ユーザが、管理対象デバイスでライン カードおよびルート プロセッサなどのすべての物理エンティティのメモリ プールの統計情報をリモートで監視できるようにする SNMP オブジェクトについて説明します。これは、特に多くのライン カードが存在する可能性があるハイエンド デバイスに有用です。最近、MIB は、バッファ プールとバッファ キャッシュ統計情報を提供するように強化されました。

MIB によって提供される統計情報に加えて、最大数のメモリ バッファが変更されたときに（つまり新しいバッファ ピークに到達）送信されるように、SNMP 通知（トラップまたはインフォーム）を設定できます。

メモリ プールをイネーブルにする方法：SNMP 通知のサポート

手順の概要

1. イネーブル化
2. **configureterminal**
3. **snmp-serverenabletrapsmemory**[**bufferpeak**]
4. **snmp-serverhost** {*hostname* | *ip-address*} [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*] [**vrf** *vrf-name*]
5. **exit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	イネーブル化 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**snmp-serverenabletrapsmemory**[**bufferpeak**] 例： `Device(config)# snmp-server enable traps memory bufferpeak`	CISCO-ENHANCED-MEMPOOL-MIB でバッファ ピーク通知（トラップまたはインフォーム）のみをイネーブルにします。
ステップ 4	**snmp-serverhost** {*hostname* | *ip-address*} [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*] [**vrf** *vrf-name*] 例： `Device(config)# snmp-server host NMS-host1.example.com community1 memory`	指定したホストへのバッファ ピーク通知の送信をイネーブルにします。

	コマンドまたはアクション	目的
ステップ 5	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

メモリ プールの設定例：SNMP 通知のサポート

メモリ プールのイネーブル化：SNMP 通知サポートの例

次の例では、使用可能なすべてのメモリ関連 SNMP 通知がコミュニティ ストリング public を使用してホスト myhost.cisco.com にインフォームとして送信されるように設定されます。

```
Device(config)# snmp-server enable traps memory bufferpeak
Device(config)# snmp-server host myhost.cisco.com informs version 3 public memory
```

このリリースからは、バッファのピーク メモリの通知タイプだけが使用可能であることに注意してください。追加メモリ通知タイプ キーワードは今後のリリースで追加される可能性があります。

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
Cisco IOS コマンド	『Cisco IOS Master Commands List, All Releases』
『Cisco IOS SNMP Support Command Reference』	『Cisco IOS SNMP Support Command Reference』

標準および RFC

標準/RFC	Title
CBC-DES（DES-56）標準	『*Symmetric Encryption Protocol*』
規格 58	『*Structure of Management Information Version 2 (SMIv2)*』＞

標準/RFC	Title
RFC 1067	『A Simple Network Management Protocol』
RFC 1091	『Telnet terminal-type option』
RFC 1098	Simple Network Management Protocol（SNMP）
RFC 1157	Simple Network Management Protocol（SNMP）
RFC 1213	『Management Information Base for Network Management of TCP/IP-based internets: MIB-II』
RFC 1215	『Convention for defining traps for use with the SNMP』
RFC 1901	『Introduction to Community-based SNMPv2』
RFC 1905	『Common Management Information Services and Protocol over TCP/IP (CMOT)』
RFC 1906	『Telnet X Display Location Option』
RFC 1908	Simple Network Management Protocol（SNMP）
RFC 2104	『HMAC: Keyed-Hashing for Message Authentication』
RFC 2206	『RSVP Management Information Base using SMIv2』
RFC 2213	『Integrated Services Management Information Base using SMIv2』
RFC 2214	『Integrated Service Management Information Base Guaranteed Service Extensions using SMIv2』
RFC 2233	『The Interface Group MIB using SMIv2』
RFC 2271	『An Architecture for Describing SNMP Management Frameworks』
RFC 2570	『Introduction to Version 3 of the Internet-standard Network Management Framework』
RFC 2578	『Structure of Management Information Version 2 (SMIv2)』
RFC 2579	『Textual Conventions for SMIv2』
RFC 2580	『Conformance Statements for SMIv2』
RFC 2981	『Event MIB』
RFC 3413	『SNMP v3 Applications』
RFC 3415	『View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)』

MIB

MIB	MIB のリンク
• Cisco SNMPv2 • イーサネット ライク インターフェイス MIB • 『Event MIB』 • デルタ、ワイルドカード、集約の式 MIB サポート • インターフェイス グループ MIB（IF-MIB） • インターフェイス グループ MIB 機能拡張 • ユニバーサル ゲートウェイおよびアクセス サーバのための MIB 機能拡張	選択したプラットフォーム、Cisco IOS XE ソフトウェア リリース、およびフィーチャ セットの MIB の場所を検索しダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

シスコのテクニカル サポート

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

メモリ プールの機能情報：SNMP 通知のサポート

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 *6*：メモリ プールの機能情報：*SNMP* 通知のサポート

機能名	リリース	機能情報
メモリ プール：SNMP 通知のサポート	12.3(4)T 12.2(22)S 12.2(33)SRA 12.2(33)SXH	この機能は Cisco 拡張メモリ プール MIB（CISCO-ENHANCED-MEMPOOL-MIB）の SNMP 通知をイネーブルにする CLI コマンドを追加します。

第 7 章

定期的な管理情報ベース（MIB）データの収集および転送のメカニズム

定期的な MIB データの収集および転送のメカニズムの機能を使用すると、Cisco IOS XE ベースのデバイスから選択した MIB データを特定のネットワーク管理システム（NMS）へ定期的に転送できます。コマンドライン インターフェイス（CLI）を使用すると、複数の MIB のデータをリストにグループ化して、ポーリング間隔（データ収集の頻度）を設定できます。リスト内の MIB オブジェクトはすべて、この指定された間隔で定期的にポーリングされます。その後、リストから収集されたデータは、TFTP、rcp、または FTP を使用して、ユーザ指定の転送間隔で指定した NMS に転送できます。

- 機能情報の確認，129 ページ
- 定期的な MIB データの収集および転送のメカニズムの前提条件，130 ページ
- 定期的な MIB データの収集および転送のメカニズムの制約事項，130 ページ
- 定期的な MIB データの収集および転送のメカニズムについて，130 ページ
- 定期的な MIB データの収集および転送のメカニズムの設定方法，132 ページ
- 定期的な MIB データの収集および転送のメカニズムの設定例，144 ページ
- その他の参考資料，148 ページ
- 定期的な MIB データの収集および転送のメカニズムの機能情報，149 ページ

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、Bug Search Tool およびご使用の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。このモジュールで説明される機能に関する情報、および各機能がサポートされるリリースの一覧については、機能情報の表を参照してください。

プラットフォームのサポートおよびシスコ 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。

定期的な MIB データの収集および転送のメカニズムの前提条件

この機能を使用するには、管理情報の簡易ネットワーク管理プロトコル（SNMP）モデルの知識が必要です。また、ネットワーク デバイスでモニタする MIB 情報や、モニタ対象である MIB オブジェクトの OID またはオブジェクト名を知っている必要があります。

定期的な MIB データの収集および転送のメカニズムの制約事項

SNMP を使用した Cisco Data Collection MIB 設定は、現在実装されていません。

特定の制約事項については、定期的な MIB データの収集および転送のメカニズムの設定方法, （132 ページ）のタスクを参照してください。

定期的な MIB データの収集および転送のメカニズムについて

（注） 定期的な MIB データの収集および転送のメカニズムはバルク統計情報機能とも呼ばれます。

SNMP のオブジェクトとインスタンス

SNMP 管理情報のタイプ（またはクラス）をオブジェクトと呼びます。管理情報のタイプの特定のインスタンスをオブジェクトインスタンス（または SNMP 変数）と呼びます。バルク統計情報収集を設定するには、バルク統計情報オブジェクトリストを使用してモニタするオブジェクトタイプと、バルク統計情報スキーマを使用して収集するオブジェクトの特定のインスタンスを指定する必要があります。

オブジェクト識別子（OID）と呼ばれる一連の番号を使用すると、MIB、MIB テーブル、MIB オブジェクト、オブジェクトのインデックスをすべて指定できます。OID は、バルク統計情報オブジェクトリスト（一般的なオブジェクト用）とバルク統計情報スキーマ（特定のオブジェクトインスタンス用）の両方のバルク統計情報収集の設定に使用されます。

バルク統計情報オブジェクト リスト

ポーリング対象の MIB オブジェクトをグループ化するには、1 つまたは複数のオブジェクト リストを作成する必要があります。バルク統計情報オブジェクトリストは、同じMIBインデックスを共有する、ユーザ指定のMIBオブジェクトのセットです。オブジェクトリストは、指定した名前によって識別されます。名前付きのバルク統計情報オブジェクト リストを使用すると、異なるバルク統計情報スキーマで同じ設定を再利用できます。

オブジェクト リストのオブジェクトはすべて、同じ MIB インデックスを共有する必要があります。ただし、オブジェクトが同じ MIB 内に存在したり、同じ MIB テーブルに属する必要はありません。たとえば、ifInOctets とファストイーサネット MIB オブジェクトを同じスキーマでグループ化することが可能です。これは、両方のオブジェクトに対して含まれているテーブルが ifIndex によって指標付けされるためです。

バルク統計情報スキーマ

定期的なMIBデータの収集および転送のメカニズムに対するデータの選択には、次の情報を含むスキーマの定義が必要です。

- オブジェクト リストの名前。
- 上記のオブジェクトリスト内のオブジェクト用に取得する必要がある（特定またはワイルドカードを使用した）インスタンス。
- 指定したインスタンスに対して必要なサンプリングの頻度（ポーリング間隔）。

バルク統計情報スキーマも、指定した名前によって識別されます。この名前は、転送オプションを設定する際に使用されます。

バルク統計情報転送オプション

収集するデータを設定した後、収集した全データを使用して単一の仮想ファイル（VFile または「バルク統計情報ファイル」）が作成されます。このファイルは、FTP、RCP、またはTFTPを使用してネットワーク管理ステーション（NMS）に転送できます。このファイルの転送頻度を指定できます。デフォルトの転送間隔は、30分に1回です。何らかの理由でプライマリ ネットワーク管理ステーションに転送できない場合に使用されるセカンダリ宛先を設定することもできます。

転送間隔の値は、ローカルのバルク統計情報ファイルの収集期間（収集間隔）でもあります。収集期間が終了すると、そのバルク統計情報ファイルは凍結し、データを格納するためにローカルのバルク統計情報ファイルが新たに作成されます。その後、凍結したバルク統計情報ファイルは指定した宛先に転送されます。

デフォルトでは、ローカルのバルク統計情報ファイルはNMSに正常に転送された後に削除されます。ただし、指定した期間だけバルク統計情報ファイルをメモリに保存するようにルーティングデバイスを設定できます。

プライマリ NMS またはセカンダリ NMS へ正常に転送されなかった場合、SNMP 通知（trap）を NMS に送信できます。また、転送に失敗した場合は、syslog メッセージがローカルデバイスに記録されます。

定期的な MIB データの収集および転送のメカニズムの利点

定期的なMIBデータの収集および転送のメカニズム（バルク統計情報機能）では、バルク ファイルMIB（CISCO-BULK-FILE-MIB.my）と同じ機能の多くを使用できますが、重要な利点がいくつかあります。

主な利点は、この機能が CLI を使用して設定でき、外部のモニタリング アプリケーションが不要なことです。

定期的なMIBデータの収集および転送のメカニズムでは、バルク統計情報ファイルの保存は主に（揮発性または永続的な）ローカル ストレージが十分にある中規模からハイエンドのプラットフォームをターゲットとしています。バルク統計情報ファイルをローカルに保存すると、一時的なネットワーク停止時にデータ損失を最小限に抑えられます。

この機能にはバルク ファイル MIB よりも強力なデータ選択機能があるため、異なるテーブルのMIBオブジェクトをデータ グループ（オブジェクトリスト）にグループ化することも可能です。また、この機能はより柔軟性のあるインスタンス選択メカニズムを備えています。このメカニズムでは、アプリケーションは MIB テーブル全体を取得するように制限されていません。

定期的な MIB データの収集および転送のメカニズムの設定方法

バルク統計情報オブジェクト リストの設定

定期的な MIB データの収集および転送のメカニズムを設定する場合の最初の手順は、1 つまたは複数のオブジェクト リストを設定することです。

（注） バルク統計情報オブジェクトリスト内のオブジェクトはすべて、同じMIBインデックスによって指標付けされる必要があります。ただし、オブジェクトリスト内のオブジェクトが同じMIBまたは MIB テーブルに属する必要はありません。

（**add** コマンドを使用した）OID ではなくオブジェクト名を指定する場合、インターフェイス MIB（IF-MIB.my）、Cisco 専用アクセス レート MIB（CISCO-CAR-MIB.my）、MPLS トラフィック エンジニアリング MIB（MPLS-TE-MIB.my）のオブジェクト名だけを使用できます。

手順の概要

1. イネーブル化
2. **configureterminal**
3. **snmpmibbulkstatobject-list***list-name*
4. **add** {*oid* | *object-name*}
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	イネーブル化 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**snmpmibbulkstatobject-list***list-name* 例： `Device(config)# snmp mib bulkstat object-list ifMib`	SNMP バルク統計情報オブジェクト リストを定義し、バルク統計情報オブジェクト リスト 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	**add** {*oid*	*object-name*}
		
		例：
		
		`Device(config-bulk-objects)# add 1.3.6.1.2.1.2.2.1.11`
		
		例：
		
		`Device(config-bulk-objects)# add ifAdminStatus`
		
		例：
		
		`Device(config-bulk-objects)# add ifDescr`
		
		例：
		
		.
			MIB オブジェクトをバルク統計情報オブジェクト リストに追加します。
			• モニタ対象の全オブジェクトがこのリストに追加されるまで、必要に応じて繰り返します。

	コマンドまたはアクション	目的
	例： . 例： .	
ステップ 5	**end** 例： `Device(config-bulk-objects)# end`	バルク統計情報オブジェクト リスト コンフィギュレーション モードを終了して、特権 EXEC モードに戻ります。

バルク統計情報スキーマの設定

定期的な MIB データの収集および転送のメカニズムを設定する場合の次の手順は、1 つまたは複数のスキーマを設定することです。

はじめる前に

スキーマで使用されるバルク統計情報オブジェクト リストを定義する必要があります。

（注） 同時にスキーマに関連付けできるオブジェクト リストは 1 つだけです。

手順の概要

1. イネーブル化
2. **configureterminal**
3. **snmpmibbulkstatschema***schema-name*
4. **object-list***list-name*
5. **instance** {**exact** | **wild**} {**interface***interface-id* [**sub-if**] | **controller***controller-id* [**sub-if**] | **oid***oid*}
6. **instancerangestart***oid***end***oid*
7. **instancerepetition***oid-instance***max***repeat-number*
8. **poll-interval***minutes*
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	イネーブル化 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**snmpmibbulkstatschema***schema-name* 例： `Device(config)# snmp mib bulkstat schema intE0`	バルク統計情報スキーマの名前を指定し、バルク統計情報スキーマ（config-bulk-sc）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	**object-list***list-name* 例： `Device(config-bulk-sc)# object-list ifMib`	このスキーマに含めるバルク統計情報オブジェクト リストを指定します。スキーマごとにオブジェクト リストを 1 つだけ指定してください （複数の **object-list** コマンドが実行されると、より新しいコマンドによって先行のコマンドが上書きされます）。

ステップ 5	**instance** {**exact** | **wild**} {**interface***interface-id* [**sub-if**] | **controller***controller-id* [**sub-if**] | **oid***oid*}
	
	例：
	`Device(config-bulk-sc)# instance wild oid 1`
	
	例：
	`Device(config-bulk-sc)# instance exact interface gigabitinterface0/0/1 sub-if`
		このスキーマにおけるオブジェクトのインスタンス情報を指定します。
		• **instanceexact** コマンドは、指定されたインスタンスが完全な OID であることを示しています（オブジェクト リストに追加されている場合）。
		• **instancewild** コマンドは、指定した OID のすべてのサブインデックスがこのスキーマに属することを示しています。**wild** キーワードを使用すると、部分的に「ワイルドカードを使用した」インスタンスを指定できます。
		• インスタンス OID を指定する代わりに、特定のインターフェイスを指定できます。**interface***interface-id* 構文を使用すると、インターフェイスの ifIndex OID を指定する代わりにインターフェイス名と番号（例：Fast Ethernet interface 0）を指定できます。同様に、**controller***controller-id* 構文を使用すると、コントローラ カードを指定できます（インターフェイス）。このオプションはプラットフォームによって異なります。

	コマンドまたはアクション	目的
		• オプションの **sub-if** キーワードには、指定したインターフェイスのすべてのサブインターフェイスに対する ifIndex が含まれます（インターフェイスまたはコントローラの指定後に追加された場合）。 • 1 つのスキーマに設定できる **instance** コマンドは 1 つだけです（複数の instance コマンドが実行されると、新しいコマンドによって以前のコマンドが上書きされます）。
ステップ 6	**instancerangestart***oid***end***oid* 例： `Device(config-bulk-sc)# instance range start 1 end 2`	（任意）**snmpmibbulkstatschema** コマンドと組み合わせて使用すると、**instancerange** コマンドは、データを収集するインスタンス範囲を設定できます。
ステップ 7	**instancerepetition***oid-instance***max***repeat-number* 例： `Device(config-bulk-sc)# instance repetition 1 max 4`	（任意）**snmpmibbulkstatschema** コマンドと組み合わせて使用すると、**instancerepetition** コマンドは、MIB オブジェクトの一定数のインスタンスに対してデータ収集を繰り返すよう設定できます。
ステップ 8	**poll-interval***minutes* 例： `Device(config-bulk-sc)# poll-interval 10`	このスキーマで指定されたオブジェクトインスタンスからデータを収集する頻度を分単位で設定します。デフォルトは、5 分に 1 回です。 有効範囲は 1 ～ 20000 です。
ステップ 9	**end** 例： `Device(config-bulk-sc)# end`	バルク統計情報スキーマコンフィギュレーションモードを終了し、特権 EXEC モードに戻ります。

バルク統計情報転送オプションの設定

定期的な MIB データの収集および転送のメカニズムを設定する場合の最後の手順は、転送オプションを設定することです。収集された MIB データは、VFile（仮想ファイル。このマニュアル内ではバルク統計情報ファイルとも呼ばれている）と呼ばれるローカル ファイルのようなエンティティに格納されます。このファイルは、ユーザが指定した間隔でリモートのネットワーク管理ステーション（NMS）に転送できます。

はじめる前に

バルク統計情報オブジェクト リストとバルク統計情報スキーマは、バルク統計情報転送オプションを設定する前に定義する必要があります。

（注） 転送を実行できるのは、schemaASCII（cdcSchemaASCII）形式を使用した場合に限ります。SchemaASCII は、データ値を解析するためのパーサー フレンドリなヒントを含む ASCII 形式です。

手順の概要

1. イネーブル化
2. **configureterminal**
3. **snmpmibbulkstattransfer***transfer-id*
4. **buffer-size***bytes*
5. **format** {**bulkBinary** | **bulkASCII** | **schemaASCII**}
6. **schema***schema-name*
7. **transfer-interval***minutes*
8. **urlprimary***url*
9. **urlsecondary***url*
10. **retry***number*
11. **retain***minutes*
12. イネーブル化
13. **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	イネーブル化 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	**snmpmibbulkstattransfer***transfer-id* 例： `Device(config)# snmp mib bulkstat transfer bulkstat1`	転送設定を名前（*transfer-id*）で識別し、バルク統計情報転送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	**buffer-size***bytes* 例： `Device(config-bulk-tr)# buffer-size 3072`	（任意）バルク統計情報データ ファイルの最大サイズをバイト単位で指定します。有効範囲は 1024 ～ 2147483647 バイトです。デフォルトのバッファ サイズは 2048 バイトです。 （注） 設定可能なバッファ サイズの制限は、安全機能としてだけ使用できます。通常のバルク統計情報ファイルは、一般的にデフォルト値を満たしたり超えてはいけません。
ステップ 5	**format** {**bulkBinary** | **bulkASCII** | **schemaASCII**} 例： `Device(config-bulk-tr)# format schemaASCII`	（任意）バルク統計情報データ ファイル（VFile）の形式を指定します。デフォルトは schemaASCII です。 （注） 転送を実行できるのは、schemaASCII（cdcSchemaASCII）形式を使用した場合に限ります。SchemaASCII は、データ値を解析するためのパーサー フレンドリなヒントを含むヒト可読形式です。

ステップ 6	**schema***schema-name*
	
	例：
	
	`Device(config-bulk-tr)# schema ATM2/0-IFMIB`
	
	例：
	
	`Device(config-bulk-tr)# schema ATM2/0-CAR`
	
	例：
	
	`Device(config-bulk-tr)# schema FastEthernet2/1-IFMIB`
	
	例：
	
	.
	
	例：
	
	.
	
	例：
	
	.
		転送するバルク統計情報スキーマを指定します。必要に応じて、このコマンドを繰り返します。複数のスキーマを単一の転送設定に関連付けることができます。収集された全データが単一のバルク データ ファイル（VFile）になります。

	コマンドまたはアクション	目的
ステップ 7	**transfer-interval***minutes* 例： `Device(config-bulk-tr)# transfer-interval 20`	（任意）バルク統計情報ファイルの転送頻度を分単位で指定します。デフォルト値は、30分に1回です。転送間隔は、収集間隔と同じです。
ステップ 8	**urlprimary***url* 例： `Device(config-bulk-tr)# url primary ftp://user:password@host/folder/bulkstat1`	バルク統計情報データ ファイルを転送するネットワーク管理システム（ホスト）と転送に使用するプロトコルを指定します。この宛先は、ユニフォーム リソース ロケータ（URL）として指定されます。 • FTP、rcp、またはTFTPは、バルク統計情報ファイルの転送に使用できます。
ステップ 9	**urlsecondary***url* 例： `Device(config-bulk-tr)# url secondary tftp://10.1.0.1/tftpboot/user/bulkstat1`	（任意）プライマリ ロケーション ファイルへの転送イベントで使用する、バックアップの転送先とプロトコルを指定します。 • FTP、rcp、またはTFTPは、バルク統計情報ファイルの転送に使用できます。
ステップ 10	**retry***number* 例： `Device(config-bulk-tr)# retry 1`	（任意）送信の再試行回数を指定します。デフォルト値は0（つまり、再試行しない）です。 • バルク統計情報ファイルを送信しようとして失敗した場合に、このコマンドを使用してファイルの再送信を試みるように設定できます。1回の再試行で、最初にプライマリ宛先への送信を試み、転送が失敗した場合は、次にセカンダリ宛先への送信を試みます。たとえば、再試行値が 1 の場合、最初にプライマリ URL への送信、次にセカンダリ URL への送信を試みた後、再びプライマリ URL への送信、次にセカンダリ URL への送信を試みます。 • 有効範囲は 0 ～ 100 です。
ステップ 11	**retain***minutes* 例： `Device(config-bulk-tr)# retain 60`	（任意）収集間隔と転送試行の設定が完了したら、バルク統計情報ファイルをシステムメモリに保存する期間を分単位で指定します。デフォルト値は 0 です • ゼロ（0）は、転送に成功するとただちにファイルが削除されることを示しています。

	コマンドまたはアクション	目的
		（注） **retry** コマンドを使用する場合、保持間隔を 0 よりも大きく設定する必要があります。再試行の間隔は、保持間隔を再試行回数で割ったものです。たとえば、**retain10** と **retry2** に設定した場合、再試行は 5 分に 1 回行われます。したがって、retain 0 に設定した場合、再試行は行われません。 • 有効範囲は 0 ～ 20000 です。
ステップ 12	イネーブル化 例： `Device(config-bulk-tr)# enable`	この設定でバルク統計情報のデータ収集と転送のプロセスを開始します。 • このアクションを正常に実行させるには、0 以外の数字が設定されたスキーマを最低 1 つは設定する必要があります。 • 定期的なデータ収集と転送の動作が開始されるのは、このコマンドが設定されている場合だけです。逆に、**noenable** コマンドが設定されていると、収集プロセスが停止します。その後ろに **enable** が設定されていると、動作が再び開始されます。 • **Enable** コマンドを使用して収集プロセスが開始されるたびに、新しいバルク統計情報ファイルにデータが収集されます。**noenable** コマンドを使用すると、収集したデータの転送プロセスがただちに開始されます（つまり、既存のバルク統計情報ファイルが指定した管理ステーションに転送される）。
ステップ 13	**end** 例： `Device(config-bulk-tr)# end`	バルク統計情報転送コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

トラブルシューティングのヒント

リブートにまたがってのインターフェイスの識別に ifAlias 値を使用することの代替方法として、Cisco 回線インターフェイス MIB（CISCO-CIRCUIT-INTERFACE-MIB.my）内の cciDescr オブジェクトを使用する方法があります。この MIB オブジェクトは、ATM インターフェイス、フレームリレーインターフェイスなどの回線ベースのインターフェイスにしか使用できません。Cisco IOS 機能の FTS-731 では Simple Network Management Protocol（SNMP）のための回線インターフェイス ID パーシステンス機能が導入されました。この機能により、回路のユーザー定義名（cciDescr

オブジェクトで定義）がリブートにまたがって維持され、回路ベース インターフェイスの一貫した識別が可能になります。

バルク統計情報収集のモニタリングのイネーブル化

任意で、SNMP 通知の送信をイネーブルにできます。これにより、定期的な MIB データの収集および転送のメカニズム（バルク統計情報機能）の転送ステータスに情報が表示されます。

手順の概要

1. イネーブル化
2. **configureterminal**
3. **snmp-servercommunity***string* [**view***view-name*] [**ro** | **rw**] [*acl-number*]
4. **snmp-serverenabletrapsbulkstat** [**collection** | **transfer**]
5. **snmp-serverhost***host-address* [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port***port*] [**bulkstat**]
6. **exit**
7. **copyrunning-configstartup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	イネーブル化 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**snmp-servercommunity***string* [**view***view-name*] [**ro**	**rw**] [*acl-number*] 例： `Device(config)# snmp-server community public`	デバイスに対して SNMP コミュニティと SNMP アクセスの各オプションを指定します。
ステップ 4	**snmp-serverenabletrapsbulkstat** [**collection**	**transfer**] 例： `Device(config)# snmp-server enable traps bulkstat`	バルク統計情報の SNMP 通知（トラップまたはインフォーム）の送信をイネーブルにします。 （CISCO-DATA-COLLECTION-MIB で定義される）次の通知はこのコマンドでイネーブルになります。

	コマンドまたはアクション	目的
		• 転送（cdcFileXferComplete）：転送試行に成功した場合と転送試行に失敗した場合に送信します（トラップ内の varbind である cdcFilXferStatus オブジェクトは、転送に成功したかどうかを示しています）。 • 収集（cdcVFileCollectionError）：データ収集が正常に実行されなかった場合に送信します。この状況の原因の 1 つには、データ収集を実行するのに十分なメモリがデバイス上になかったことが考えられます。
ステップ 5	**snmp-serverhost***host-address* [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port***port*] [**bulkstat**] 例： `Device(config)# snmp-server host informs public bulkstat`	SNMP 通知の受信者（ホスト）と追加の転送オプションを指定します。
ステップ 6	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了します。
ステップ 7	**copyrunning-configstartup-config** 例： `Device# copy running-config startup-config`	（任意）現在の設定をスタートアップ コンフィギュレーション ファイルとして NVRAM に保存します。

定期的な MIB データの収集および転送のメカニズムのモニタリングとトラブルシューティング

この機能に対する **show** コマンドでは、バルク統計情報プロセスのステータスが表示されます。**debug** コマンドを使用すると、テクニカル サポート目的でデバッグ メッセージの標準セットがイネーブルになります。

手順の概要

1. イネーブル化
2. **showsnmpmibbulkstattransfer** [*transfer-name*]
3. **debugsnmpbulkstat**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	イネーブル化 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。

ステップ 2	**showsnmpmibbulkstattransfer** [*transfer-name*]
	
	例：
	
	`Device# show snmp mib bulkstat transfer`
	
	`Transfer Name : ifmib`
	`Retained files`
	`File Name : Time Left (in seconds) :STATE`
	`----------`
	`ifmib_Router_020421_100554683 : 173 : Retry (2 Retry attempt(s) Left)`
	`ifmib_Router_020421_100554683 : 53 : Retained`
		（任意）この機能に対する **show** コマンドでは、データ収集を完了したシステム上のすべてのバルク統計情報の仮想ファイル（VFile）が一覧表示されます（収集が完了していないデータ ファイルは表示されません）。
		
		収集が完了したローカルのすべてのバルク統計情報ファイル、バルク統計情報ファイルが削除されるまでの残り時間（残りの保持期間）、バルク統計情報ファイルの状態が出力に一覧表示されます。
		
		バルク統計情報ファイルの「状態」は次のいずれかになります。
		
		• Queued：このバルク統計情報ファイルのデータ収集が完了し（つまり、転送間隔に達した）、バルク統計情報ファイルが設定された宛先への転送を待機していることを示しています。
		
		• Retry：転送試行に 1 回または複数回失敗し、ファイル転送が再試行されることを示しています。残りの再試行回数がカッコ内に表示されます。
		
		• Retained：バルク統計情報ファイルが正常に転送されたか、または設定した回数の再試行が完了したことを示しています。

	コマンドまたはアクション	目的
		ヒント 転送に成功したかどうかを判断するには、バルク統計情報の SNMP 通知をイネーブルにします。 （設定したすべての転送ではなく）名前付きの転送の状態だけを表示するには、*transfer-name* 引数で転送名を指定します。
ステップ 3	**debugsnmpbulkstat** 例： `Device# debug snmp bulkstat`	（任意）バルク統計情報機能に対して標準のデバッグ出力をイネーブルにします。デバッグ出力には、バルク統計情報ファイルの作成、転送、削除に関するメッセージが含まれています。

定期的な MIB データの収集および転送のメカニズムの設定例

定期的な MIB データの収集および転送のメカニズムの設定の例

このセクションでは、定期的な MIB データの収集および転送のメカニズム（バルク統計情報機能）のすべての設定例を示します。この例については、次のサブ項目で説明しています。

転送パラメータ

「定期的な MIB データの収集および転送のメカニズムの設定」の例には、次の転送パラメータが使用されています。

- 転送間隔（収集間隔）：30 分
- プライマリ URL：ftp://john:pswrd@cbin2-host/users/john/bulkstat1
- セカンダリ URL：tftp://john@10.1.1.1/tftpboot/john/bulkstat1
- 転送形式：schemaASCII
- 再試行間隔：6 分後に再試行（retry = 5、retain = 30、30 分の保持間隔に 5 回の再試行）

ポーリングの要件

「定期的な MIB データの収集および転送のメカニズムの設定」の例には、ATM インターフェイス 2/0 およびファスト イーサネット インターフェイス 2/1 に対する次のポーリングの要件が使用されています。

ATM インターフェイス 2/0

- ポーリング対象のオブジェクト：ifInOctets、ifOutOctets、ifInUcastPkts、ifInDiscards、CcarStatSwitchedPkts、CcarStatSwitchedBytes、CcarStatFilteredBytes
- ポーリング間隔：5 分に 1 回
- インスタンス：メイン インターフェイスとすべてのサブインターフェイス
- CAR MIB オブジェクトの場合、指定されたインターフェイスに関連するすべてのインスタンスをポーリングします。

ファスト イーサネット インターフェイス 2/1

- ポーリング対象のオブジェクト：ifInOctets、ifOutOctets、ifInUcastPkts、ifInDiscards、CcarStatSwitchedPkts、CcarStatSwitchedBytes、CcarStatFilteredBytes
- ポーリング間隔：10 分に 1 回
- インスタンス：メイン インターフェイスだけがモニタ対象になります。
- CAR MIB オブジェクトの場合、（メイン インターフェイス上での）着信方向のパケットに関するインスタンスだけが含まれます。

オブジェクト リストの設定

IF-MIB オブジェクトと CAR-MIB オブジェクトは同じインデックスを持たないため、異なるスキーマの一部として存在する必要があることに注意してください。ただし、ATM インターフェイスとファストイーサネットインターフェイスに必要なオブジェクトは同じであるため、オブジェクト リストは各スキーマに再利用できます。したがって、次の例では、IF-MIB オブジェクト用にオブジェクト リストが作成され、CAR-MIB オブジェクト用に別のオブジェクト リストが作成されます。

```
snmp mib bulkstat object-list ifmib
add ifInoctets
add ifOutoctets
add ifInUcastPkts
add ifInDiscards
exit
snmp mib bulkstat object-list CAR-mib
add CcarStatSwitchedPkts
add CcarStatSwitchedBytes
add CcarStatFilteredBytes
exit
```

スキーマ定義の設定

次のバルク統計情報スキーマの設定では、各インターフェイスに 2 つのスキーマ（IF-MIB オブジェクト インスタンス用と CAR-MIB オブジェクト インスタンス用）が定義されています。

```
! ATM IF-MIB schema
snmp mib bulkstat schema ATM2/0-IFMIB
! The following command points to the IF-MIB object list, defined above.
object-list ifmib
poll-interval 5
instance exact interface ATM2/0 subif
exit
! ATM CAR-MIB schema
snmp mib bulkstat schema-def ATM2/0-CAR
object-list CAR-mib
poll-interval 5
instance wildcard interface ATM2/0 subif
exit
!FastEthernet IF-MIB schema
snmp mib bulkstat schema FastEthernet2/1-IFMIB
object-list ifmib
poll-interval 5
instance exact interface FastEthernet2/1
exit
! FastEthernet CAR-MIB schema
snmp mib bulkstat schema FastEthernet2/1-CAR
object-list CAR-mib
poll-interval 5
! Note: ifindex of FastEthernet2/1 is 3
instance wildcard oid 3.1
exit
```

転送パラメータの設定

バルク統計情報ファイルの転送には、転送設定に bulkstat1 という名前が設定されます。4 つのスキーマ定義にはすべて、次の転送設定が含まれています。

```
snmp mib bulkstat transfer bulkstat1
schema ATM2/0-IFMIB
schema ATM2/0-CAR
schema FastEthernet2/1-IFMIB
schema FastEthernet2/1-CAR
url primary ftp://username1:pswrd@cbin2-host/users/username1/bulkstat1
url secondary tftp://username1@10.1.0.1/tftpboot/username1/bulkstat1
format schemaASCII
transfer-interval 30
retry 5
buffer-size 1024
retain 30
end
copy running-config startup-config
```

ステータスの表示

次の **show snmp mib bulkstat transfer** コマンドの出力例は、最新ファイルに対する最初の転送試行と最初の再試行が失敗し、追加で 4 回の再試行が行われることを示しています。

```
Device# show snmp mib bulkstat transfer

Transfer Name : bulkstat1
```

```
Primary URL ftp://user:XXXXXXXX@192.168.200.162/
Secondary ftp://user:XXXXXXXX@192.168.200.163/
      Retained files

      File Name              : Time Left (in seconds)     : STATE
     ---------------------------------------------------------------------
      bulkstat1_Router_030307_102519739: 1196         :Retry(4 Retry attempt(s) Left)
      bulkstat1_Router_030307_102219739: 1016         :Retained
      bulkstat1_Router_030307_101919739:  836         :Retained
```

バルク統計情報ファイルのファイル名は、**url** コマンドで指定した名前を次のように拡張して生成されます。

specified-filename _device-name _date_time-stamp

デバイス名は、CLI プロンプトで指定した送信元デバイスの名前です。

タイム スタンプ形式はシステム設定によって異なります。通常、日付形式は YYYYMMDD または YYMMDD です。タイム スタンプには 24 時間制を使用し、形式は HHMMSSmmm（mmm はミリ秒）です。

上記の例では、2003 年 3 月 7 日の午前 10:25、午前 10:22、午前 10:19 にファイルが作成されています。

バルク統計情報出力ファイル

転送先で受信するバルク統計情報ファイルに表示される出力例は次のとおりです。この出力では、バルク統計情報ファイルの名前は bulkstat1_Router_20030131_193354234 です。また、設定が変更されてスキーマ Fast Ethernet2/1-IFMIB のスキーマ定義（Schema-def）がファイルに追加されていることに注意してください（「!」で示されているコメント行を参照してください）。

```
Schema-def  ATM2/0-IFMIB "%u, %s,  %u, %u, %u, %u"
epochtime ifDescr instanceoid ifInOctets ifOutOctets ifInUcastPkts ifInDiscards
Schema-def  ATM2/0-CAR  "%u, %s,  %s, %u, %u, %u, %u "
epochtime ifDescr instanceoid CcarStatSwitchedPkts  ccarStatSwitchedBytes CcarStatSwitchedPkts
  ccarStatSwitchedBytes
Schema-def  FastEthernet2/1-IFMIB "%u, %u,  %u, %u, %u, %u"
epochtime ifDescr instanceoid ifInOctets ifOutOctets ifInUcastPkts ifInDiscards
Schema-def  FastEthernet2/1-CAR  "%u, %s, %u, %u, %u, %u "
Epochtime instanceoid CcarStatSwitchedPkts  ccarStatSwitchedBytes CcarStatSwitchedPkts
ccarStatSwitchedBytes
Schema-def GLOBAL "%s, %s,  %s, %u, %u, %u, %u"
            hostname data timeofday sysuptime cpu5min cpu1min cpu5sec
ATM2/0-IFMIB: 954417080, ATM2/0, 2, 95678, 23456, 234, 3456
ATM2/0-IFMIB: 954417080, ATM2/0.1, 8, 95458, 54356, 245, 454
ATM2/0-IFMIB: 954417080, ATM2/0.2, 9, 45678, 8756, 934, 36756
ATM2/0-CAR: 954417083, ATM2/0, 2.1.1, 234, 345, 123, 124
ATM2/0-CAR: 954417083, ATM2/0, 2.2.1, 452, 67, 132, 145
ATM2/0-CAR: 954417083, ATM2/0.1, 8.1.1, 224, 765, 324 234
ATM2/0-CAR: 954417083, ATM2/0.1, 8.2.1, 234, 345, 123, 124
ATM2/0-CAR: 954417083, ATM2/0.2, 9.1.1, 234, 345, 123, 124
ATM2/0-CAR: 954417083, ATM2/0.2, 9.2.1, 452, 67, 132, 145
FastEthernet2/1-IFMIB: 954417090, FastEthernet2/1, 3, 45678, 8756, 934, 36756
FastEthernet2/1-CAR: 954417093, 3.1.1, 234, 345, 123, 124
FastEthernet2/1-CAR: 954417093, 3.1.2, 134, 475, 155, 187
ATM2/0-IFMIB: 954417100, ATM2/0, 2, 95678, 23456, 234, 3456
ATM2/0-IFMIB: 954417101, ATM2/0.1, 8, 95458, 54356, 245, 454
ATM2/0-IFMIB: 954417102, ATM2/0.2, 9, 45678, 8756, 934, 36756
ATM2/0-CAR: 954417106, ATM2/0, 2.1.1, 234, 345, 123, 124
ATM2/0-CAR: 954417107, ATM2/0, 2.2.1, 452, 67, 132, 145
ATM2/0-CAR: 954417107, ATM2/0.1, 8.1.1, 224, 765, 324 234
ATM2/0-CAR: 954417108, ATM2/0.1, 8.2.1, 234, 345, 123, 124
ATM2/0-CAR: 954417113, ATM2/0.2, 9.1.1, 234, 345, 123, 124
ATM2/0-CAR: 954417114, ATM2/0.2, 9.2.1, 452, 67, 132, 145
```

```
! Here the Schema-def for "Ehternet2/1-IFMIB" was changed on the originating device.
Schema-def  FastEthernet2/1-IFMIB "%u, %u,  %u, %u, %u, %u"
! The object ifOutDiscards has been added to the object list for this schema.

epochtime ifDescr instanceoid ifInOctets ifOutOctets ifInUcastPkts ifInDiscards
          ifOutDiscards
! The following data sample reflects the change in the configuration.
FastEthernet2/1-IFMIB: 954417090, FastEthernet2/1, 3, 45678, 8756, 934, 36756, 123
FastEthernet2/1-CAR: 954417093, 3.1.1, 234, 345, 123, 124
FastEthernet2/1-CAR: 954417093, 3.1.2, 134, 475, 155, 187
GLOBAL: Govinda, 20020129, 115131, 78337, 783337, 2%, 0%, 62%
```

その他の参考資料

ここでは、定期的なMIBデータの収集および転送のメカニズムに関連する参考資料について説明します。

関連資料

関連項目	マニュアル タイトル
Cisco IOS コマンド	『Cisco IOS Master Command List, All Releases』
SNMP コマンド	『Cisco IOS SNMP Support Command Reference』
SNMP の設定作業	『*Cisco IOS XE Network Management Configuration Guide*』の「Configuring SNMP Support」モジュール

標準および RFC

RFC	Title
なし	—

MIB

MIB	MIB のリンク
この機能は、シスコ製品に実装されているすべての MIB をサポートします。 この機能は、データの収集および転送時にエラーおよび統計情報を報告する Cisco Data Collection MIB（CISCO-DATA-COLLECTION-MIB.my）機能を使用します。 また、Cisco Data Collection MIB は、SNMP だけでなく CLI を使用したデータ収集の設定をサポートします。	選択したプラットフォーム、Cisco IOS XE ソフトウェア リリース、およびフィーチャ セットの MIB の場所を検索しダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

シスコのテクニカル サポート

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

定期的な MIB データの収集および転送のメカニズムの機能情報

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 *7*：定期的な *MIB* データの収集および転送のメカニズムの機能情報

機能名	リリース	機能情報
CISCO-DATA-COLLECTION-MIB	Cisco IOS XE Release 2.1	定期的な MIB データの収集および転送のメカニズムの機能を使用すると、Cisco IOS XE ベースのデバイスから選択した MIB データを特定のネットワーク管理システム（NMS）へ定期的に転送できます。 この機能により、次のコマンドが導入または変更されました。 **add(bulkstatobject)**、**buffer-size(bulkstat)**、**debugsnmpbulkstat**、**enable(bulkstat)**、**format(bulkstat)**、**instance(MIB)**、**instancerange**、**instancerepetition**、**object-list**、**poll-interval**、**retain**、**retry(bulkstat)**、**schema**、**showsnmpmibbulkstattransfer**、**snmpmibbulkstatobject-list**、**snmpmibbulkstatschema**、**snmpmibbulkstattransfer**、**snmp-serverenabletrapsbulkstat**、**transfer-interval**、**url(bulkstat)**。