簡易ネットワーク管理プロトコルの設定

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 では、ネットワーク内のデバイスのモニタリングと管理に使用する標準フレームワークと共通言語が提供されます。

ここでは、Cisco IOS XR ネットワーク上において SNMP の実装に必要な作業について説明します。

SNMP の実装の前提条件

適切なタスク ID を含むタスク グループに関連付けられているユーザ グループに属している必要があります。このコマンド リファレンスには、各コマンドに必要なタスク ID が含まれます。ユーザ グループの割り当てが原因でコマンドを使用できないと考えられる場合、AAA 管理者に連絡してください。

Cisco IOS XR ソフトウェアでの SNMP の使用に関する制約事項

SNMP 出力は、32 ビット幅しかありません。そのため、2^{32} を超える情報は表示できません。2^{32} は 4.29 ギガビットになります。

（注） 10 ギガビット インターフェイスは 2^{32} を超えているため、インターフェイスに関する速度情報を表示しようとすると、結果が連結形式で表示される場合があります。

10 ギガビットを超えるインターフェイスの正しい速度を表示するには、ifHighSpeed を使用できます。

SNMP の実装について

SNMP を実装するには、この項の内容を理解しておく必要があります。

SNMP 機能の概要

SNMP フレームワークは 3 つの部分で構成されます。

- SNMP マネージャ
- SNMP エージェント
- 管理情報ベース（MIB）

SNMP マネージャ

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ために使用されるシステムです。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）と呼ばれます。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。さまざまなネットワーク管理アプリケーションがSNMPとともに使用可能です。簡単なコマンドライン アプリケーションから機能が豊富なグラフィカル ユーザ インターフェイス（CiscoWorks 2000 製品ラインなど）まで、このような機能は多岐にわたっています。

SNMP エージェント

SNMPエージェントは、管理対象デバイスの内部で動作するソフトウェア コンポーネントであり、デバイスのデータを保持し、必要に応じて管理システムにそれらのデータを報告します。エージェントおよびMIB は、ルータに常駐します。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。

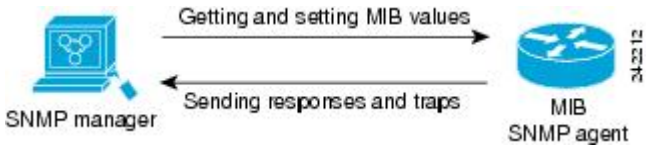
MIB

管理情報ベース（MIB）は、ネットワーク管理情報用の仮想情報ストレージ領域であり、管理対象オブジェクトの集合で構成されます。MIB 内には、MIB モジュールで定義された関連オブジェクトの集合体があります。MIB モジュールは、STD 58、RFC 2578、RFC 2579、および RFC 2580 の定義に従って、SNMP MIB モジュール言語で記述されます。なお、個々の MIB モジュールも MIB と呼ばれます。たとえば、インターフェイス グループ MIB（IF-MIB）はシステム上の MIB 内の MIB モジュールです。

SNMP エージェントには、SNMP マネージャが Get 操作や Set 操作を通じて値を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 データの保存場所である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。

次の図に、SNMPマネージャとSNMPエージェントの間の通信の関係を示します。マネージャは、MIB値の取得および設定の要求をエージェントに送信できます。エージェントはこれらの要求に応答できます。このやりとりとは別に、エージェント側からは、任意の通知（トラップ）をマネージャに送信して、ネットワークの状況をマネージャに通知できます。

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



IP-MIB のサポート

RFC4293 IP-MIB は、IPv4 と IPv6 の統計情報を個別に提供するように特別に設計されました。RFC 4293 で定義されている **ipIfStatsTable** には、インターフェイス固有の統計情報がリストされています。ipIfStatsTable の IPv6 統計情報のサポートは以前に追加されていますが、IP-MIB の IOS-XR 実装では、以前のリリースの場合 RFC4293 に従い IPv4 統計情報をサポートしていませんでした。

リリース 6.3.2 以降から、IP-MIB の IOS-XR 実装では、RFC4293 に従い IPv4 統計情報がサポートされています。これにより、インターフェイスごとに IPV4 と IPv6 の統計情報を個別に収集することができます。**ipIfStatsTable** は、2 つの**サブ ID アドレス タイプ（IPv4 または IPv6）**と**インターフェイス ifindex[1]** によってインデックス付けされます。IPv4 および IPv6 への IP-MIB サポートの実装は、読みやすさと保守性を向上させるためにリリース 6.3.2 から分離されています。

IPv4 統計情報について **ipIfStatsTable** に追加された OID のリストは次のとおりです。

- ipIfStatsInReceives
- ipIfStatsHCInReceives
- ipIfStatsInOctets
- ipIfStatsHCInOctets
- ipIfStatsOutTransmits
- ipIfStatsHCOutTransmits
- ipIfStatsOutOctets
- ipIfStatsHCOutOctets
- ipIfStatsDiscontinuityTime

iPv4 統計情報用に追加された新しい OID のリストについては、「SNMP OID Navigator」を参照してください。

SNMP バージョン

Cisco IOS XR 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。

- 簡易ネットワーク管理プロトコル バージョン 1（SNMPv1）
- 簡易ネットワーク管理プロトコル バージョン 2c（SNMPv2c）
- 簡易ネットワーク管理プロトコル バージョン 3（SNMPv3）

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。エージェントの MIB にアクセスできるマネージャのコミュニティが、IP アドレス アクセス コントロール リストおよびパスワードによって定義されます。

SNMPv2c サポートには、バルク取得メカニズム、および管理ステーションに対するより詳細なエラー メッセージ報告が含まれています。バルク取得メカニズムは、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 ではエラー処理のサポートが改善されました。たとえば、異なる種類のエラー条件が区別されるように、エラー コードが拡張されました。SNMPv1 では、これらの条件は単一のエラー コードを使用して報告されていました。エラー リターン コードでエラー タイプが報告されるようになりました。no such object exceptions、no such instance exceptions、および end of MIB view exceptions の 3 種類の例外も報告されます。

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が属するグループに合わせて設定される認証方式です。セキュリティ レベルとは、セキュリティ モデル内で許可されるセキュリティのレベルです。セキュリティ モデルとセキュリティ レベルの組み合わせによって、SNMP パケットの処理時に採用されるセキュリティ メカニズムが決まります。SNMPv3 で使用可能なセキュリティ レベルのリストについては、SNMPv1、SNMPv2、SNMPv3 のセキュリティ モデルおよびセキュリティ レベル （5 ページ） を参照してください。SNMPv3 機能は、RFC 3411 ～ 3418 をサポートします。

SNMP エージェントは、管理ステーションでサポートされる SNMP のバージョンを使用するように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。このため、1 つの管理ステーションとは SNMPv1 プロトコルを使用して通信し、1 つの管理ステーションとは SNMPv2c プロトコルを使用して通信し、もう 1 つの管理ステーションとは SMNPv3 を使用して通信することがサポートされるように、Cisco IOS-XR ソフトウェアを設定できます。

SNMPv1、SNMPv2c、および SNMPv3 の比較

SNMP v1、v2c、および v3 はすべて次の動作をサポートします。

- get-request：特定の変数から値を取得します。
- get-next-request：指定した変数の次の値を取得します。この動作はテーブル内からの変数取得によく使用されます。この動作では、SNMP マネージャに正確な変数名を認識させる必要はありません。SNMP マネージャは、必要な変数を MIB 内で順番に検索していきます。
- get-response：NMS によって送信された get-request、get-next-request、および set-request に応答する動作です。
- set-request：特定の変数に値を保存する動作です。

- trap：何らかのイベントが発生したときに、SNMP エージェントによって SNMP マネージャに送信される非送信請求メッセージです。

次の表では、SNMP v1、v2c、および v3 でサポートされるその他の主要な SNMP 機能を示します。

表 *1 : SNMPv1*、*v2c*、および *v3* 機能のサポート

機能	SNMP v1	SNMP v2c	SNMP v3
Get-Bulk 動作	×	○	○
Inform 動作	×	○（Cisco IOS XR ソフトウェアでは ×）	○（Cisco IOS XR ソフトウェアでは ×）
64 ビット カウンタ	×	○	○
テキストの表記法	×	○	○
認証	×	×	○
プライバシー（暗号化）	×	×	○
認証およびアクセス コントロール（ビュー）	×	×	○

SNMPv1、SNMPv2、SNMPv3 のセキュリティ モデルおよびセキュリティ レベル

セキュリティ レベルは、SNMP メッセージを開示から保護する必要があるかどうか、およびメッセージを認証するかどうか判断します。セキュリティ モデル内のさまざまなセキュリティ レベルは、次のとおりです。

- noAuthNoPriv：認証または暗号化を実行しないセキュリティ レベル。
- authNoPriv：認証は実行するが、暗号化を実行しないセキュリティ レベル。
- authPriv：認証と暗号化両方を実行するセキュリティ レベル。

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。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 メッセージの処理中に適用されるセキュリティ メカニズムが決まります。

次の表で、セキュリティ モデルとレベルの組み合わせについて説明します。

表 *2 : 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	コミュニティ ストリング	未対応	コミュニティ ストリングの照合を使用して認証します。

モデル	レベル	認証	暗号化	結果
v2c	noAuthNoPriv	コミュニティ ストリング	未対応	コミュニティストリングの照合を使用して認証します。
v3	noAuthNoPriv	ユーザ名	未対応	ユーザ名の照合を使用して認証します。
v3	authNoPriv	HMAC-MD5 または HMAC-SHA	非対応	HMAC[1]-MD5 [2] アルゴリズムまたは HMAC-SHA [3]に基づいて認証します。
v3	authPriv	HMAC-MD5 または HMAC-SHA	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[4] 56 ビット暗号化、および CBC[5] DES（DES-56）標準に基づいた認証を提供します。
v3	authPriv	HMAC-MD5 または HMAC-SHA	3DES	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。168 ビットの 3DES[6] レベルの暗号化を提供します。
v3	authPriv	HMAC-MD5 または HMAC-SHA	AES	HMAC-MD5 アルゴリズムまたは HMAC-SHA アルゴリズムに基づいて認証します。128 ビットの AES[7] レベルの暗号化を提供します。

1 Hash-Based Message Authentication Code
2 メッセージ ダイジェスト 5
3 セキュア ハッシュ アルゴリズム
4 データ暗号規格
5 暗号ブロック連鎖
6 Triple Data Encryption Standard
7 Advanced Encryption Standard

3DESおよびAES暗号化規格を使用するため、セキュリティ パッケージ（k9sec）がインストールされている必要があります。ソフトウェア パッケージのインストールの詳細については、『*Upgrading and Managing Cisco IOS XR Software*』を参照してください。

SNMPv3 の利点

SNMPv3は、認証、暗号化、およびアクセス コントロールを提供することで、デバイスへの安全なアクセスを実現します。これらのセキュリティの利点が追加されたことより、次のセキュリティ上の脅威に対して SNMP がセキュリティ保護されます。

- マスカレード：SNMP ユーザが別の SNMP ユーザのアイデンティティを装って、その SNMP ユーザが許可されていない管理操作を実行する脅威。

- メッセージストリームの改変：メッセージが悪意を持って並べ替え、遅延、または再生されて（サブネットワーク サービスの通常の操作によって発生するよりも大きい程度に）、SNMP が不正な管理操作を実行するようになる脅威。
- 暴露：SNMP エンジン間でのやり取りが傍受される可能性がある脅威。ローカル ポリシーの問題としてこの脅威から保護が必要な場合があります。

さらに、SNMPv3 では、SNMP 管理対象オブジェクト上のプロトコル操作に対するアクセス制御も提供されます。

SNMPv3 のコスト

SNMPv3 の認証および暗号化は、MIB オブジェクトに対する SNMP 操作の実行時の応答時間をわずかに増加させる要因となります。このコストは、SNMPv3 がもたらすセキュリティ上の利点からすれば、無視できる程度のものです。

次の表に、セキュリティ モデルとセキュリティ レベルのさまざまな組み合わせを応答時間の短い順に示します。

表 *3:* 応答時間の短い順

セキュリティ モデル	セキュリティ レベル
SNMPv2c	noAuthNoPriv
SNMPv3	noAuthNoPriv
SNMPv3	authNoPriv
SNMPv3	authPriv

ユーザベースのセキュリティ モデル

SNMPv3 ユーザベース セキュリティ モデル（USM）は SNMP メッセージレベル セキュリティを参照し、次のサービスを提供します。

- メッセージの完全性：メッセージが不正な方法で変更または破壊されず、データ シーケンスが悪意なく起こり得る範囲を超えて変更されていないことを保証します。
- メッセージ発信元の認証：受信データを発信したユーザのアイデンティティが確認されたことを保証します。
- メッセージの機密性：情報が使用不可であること、または不正なユーザ、エンティティ、またはプロセスに開示されないことを保証します。

SNMPv3 は、設定済みユーザによる管理動作のみを許可し、SNMP メッセージを暗号化します。

USM では、次の 2 つの認証プロトコルが使用されます。

- HMAC-MD5-96 認証プロトコル
- HMAC-SHA-96 認証プロトコル

USM は、メッセージ暗号化用のプライバシー プロトコルとして暗号ブロック連鎖（CBC）-DES（DES-56）を使用します。

View-Based Access Control Model

SNMP ユーザは、View-Based Access Control Model（VACM）を使用して、SNMP オブジェクトに対する読み取りアクセス、書き込みアクセス、または通知アクセスを指定することにより、SNMP 管理対象オブジェクトへのアクセスを制御できます。これは、ビューによって制限されているオブジェクトへのアクセスを防止します。これらのアクセス ポリシーは、**snmp-servergroup** コマンドでユーザ グループを設定するときに設定できます。

MIB ビュー

セキュリティ上の理由から、一部のグループのアクセスを、管理ドメイン内の一部の管理情報のみに限定できることが頻繁に重要になります。この機能を実現するために、管理オブジェクトへのアクセスは、MIB ビューによって制御されます。このビューには、表示可能な管理対象オブジェクト タイプ（およびオプションとしてオブジェクト タイプの特定のインスタンス）のセットが含まれます。

アクセス ポリシー

アクセスポリシーによって、グループのアクセス権限が決定します。アクセス権限には、次の 3 種類があります。

- 読み取りビュー アクセス：オブジェクト読み取り時に、グループに許可されているオブジェクト インスタンスのセット。
- 書き込みビュー アクセス：オブジェクト書き込み時に、グループに許可されているオブジェクト インスタンスのセット。
- 通知ビューアクセス：オブジェクトの通知での送信時に、グループに許可されているオブジェクト インスタンスのセット。

SNMP の IP precedence および DSCP サポート

SNMP による IP precedence および差分化サービスコードポイント（DSCP; DiffServ コード ポイント）のサポートでは、SNMP トラフィックに特定した QoS を提供します。ユーザがプライオリティの設定を変更することができるため、ルータで生成した SNMP トラフィックを特定の QoS クラスに割り当てます。IP precedence または IP DSCP のコード ポイント値は、パケットを重み付けランダム早期検出（WRED）でどのように処理するかを決定するのに使用します。

ルータで生成された SNMP トラフィックに IP precedence または IP DSCP が設定されると、同じルータの種類の異なる SNMP トラフィックに異なる QoS クラスを割り当てられなくなります。

IP precedence 値は、IP ヘッダーの ToS（タイプ オブ サービス）バイトの最初の 3 ビットです。IP DSCP コード ポイント値は、差分化サービス（DiffServ フィールド）バイトの最初の 6 ビットです。最大 8 つの異なる IP precedence マーキングまたは 64 の異なる IP DSCP マーキングを設定できます。

サブスクライバセッションでのセッションMIBのサポート

SNMP モニタリングでは、すべてのタイプのサブスクライバに関する情報が必要です。CISCO-SUBSCRIBER-SESSION-MIB は、サブスクライバごとのデータと集約サブスクライバ（PPPoE）データをモデル化するために定義されます。設定されたしきい値を超える集約セッション数に関する通知（トラップ）をサポートする必要があります。
CISCO-SUBSCRIBER-SESSION-MIB の汎用 MIB データ コレクタ マネージャ（DCM）のサポートにより、データ収集が高速化し、並列データの処理も向上します。

SNMP 通知

SNMP の重要な機能の 1 つは、SNMP エージェントから通知を生成できることです。これらの通知では、要求を SNMP マネージャから送信する必要はありません。Cisco IOS XR ソフトウェアでは、任意（非同期）の通知は、トラップとしてのみ生成できます。トラップは、ネットワーク上のある状態を SNMP マネージャに通知するメッセージです。通知は、不正なユーザ認証、再起動、接続の切断、隣接ルータとの接続の切断、その他の重要なイベントを表示します。

（注） インフォーム要求（インフォーム操作）はCisco IOS XR ソフトウェア ではサポートされていません。

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。送信側は、トラップが受信されたかどうかを判断できません。インフォーム要求を受信する SNMP マネージャは、SNMP 応答プロトコルデータ ユニット（PDU）でメッセージの受信を確認応答します。マネージャがインフォーム要求を受信しなかった場合、応答は返されません。送信側が応答を受信しない場合、インフォーム要求を再び送信できます。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。

ただし、インフォームはルータやネットワークのリソースをより多く消費するので、多くの場合、トラップの方が好んで使用されます。送信と同時に廃棄されるトラップと異なり、インフォーム要求は応答を受信するまで、または要求がタイムアウトになるまで、メモリ内に保持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。再送信の回数が増えるとトラフィックが増加し、ネットワークのオーバーヘッドが高くなる原因にもなります。このように、トラップとインフォーム要求の間には、信頼性とリソースのトレードオフの関係があります。

図 *2 : SNMP* マネージャで受信したトラップ

この図では、エージェント ルータは SNMP マネージャにトラップを送信します。マネージャはトラップを受信しますが、エージェントに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。

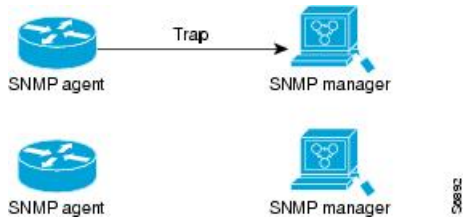
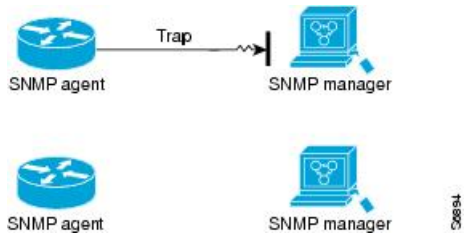


図 *3 : SNMP* マネージャで受信されなかったトラップ

次の図では、エージェントがマネージャにトラップを送信しますが、トラップはマネージャに届きません。トラップが宛先に到達しなかったことをエージェントが確認する方法がないため、トラップは再度送信されません。そのため、マネージャはこのトラップを受信できません。



セッション タイプ

サポートされているセッション タイプは次のとおりです。

- PPPoE
- IP SUB PKT
- IP SUB DHCP

Cisco IOS XR ソフトウェアでの SNMP の実装方法

ここでは、SNMP の実装方法について説明します。

snmp-server コマンドは、デフォルトで、管理イーサネット インターフェイスで SNMP をイネーブルにします。その他の帯域内インターフェイスで SNMP サーバ サポートをイネーブルにするには、『*System Security Configuration Guide for Cisco NCS 540 Series Routers*』の「*Implementing Management Plane Protection on Cisco IOS XR Software*」モジュールを参照してください。

SNMPv3 の設定

このタスクでは、ネットワーク管理およびモニタリングに SNMPv3 を設定する方法について説明します。

（注） 特定のコマンドでSNMPv3をイネーブルにすることはできません。SNMPv3は、最初に実行する **snmp-server** グローバル コンフィギュレーション コマンド（config）によってイネーブルになります。したがって、ここで **snmp-server** コマンドを実行する順序は重要ではありません。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	（任意） **snmp-serverengineidlocal** *engine-id* 例： `RP/0/RP0/CPU0:router# snmp-server engineID local 00:00:00:09:00:00:00:a1:61:6c:20:61`	ローカル SNMP エンジンの識別番号を指定します。
ステップ 3	**snmp-serverview** *view-name oid-tree* {**included** | **excluded**} 例： `RP/0/RP0/CPU0:router# snmp-server view view_name 1.3.6.1.2.1.1.5 included`	ビュー レコードを作成または変更します。
ステップ 4	**snmp-servergroup** *name* {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}} [**read** *view*] [**write** *view*] [**notify** *view*] [*access-list-name*] 例： `RP/0/RP0/CPU0:router# snmp-server group group_name v3 noauth read view_name1 write view_name2`	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。
ステップ 5	**snmp-server user** *username groupname* {**v1** | **v2c** | **v3** [**auth** {**md5** | **sha**} {**clear** | **encrypted**} *auth-password* [**privdes56** {**clear** | **encrypted**} *priv-password*]]} [*access-list-name*] 例： `RP/0/RP0/CPU0:router# snmp-server user noauthuser group_name v3`	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
ステップ 6	**commit**	
ステップ 7	（任意） **showsnmp** 例： `RP/0/RP0/CPU0:router# show snmp`	SNMP のステータスに関する情報を表示します。
ステップ 8	（任意） **showsnmpengineid** 例： `RP/0/RP0/CPU0:router# show snmp engineid`	ローカル SNMP エンジンに関する情報を表示します。
ステップ 9	（任意） **showsnmpgroup** 例： `RP/0/RP0/CPU0:router# 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。
ステップ 10	（任意） **showsnmpusers** 例： `RP/0/RP0/CPU0:router# show snmp users`	SNMP ユーザ テーブルの各 SNMP ユーザ名に関する情報を表示します。
ステップ 11	（任意） **showsnmpview** 例： `RP/0/RP0/CPU0:router# show snmp view`	関連する MIB ビュー ファミリー名、ストレージタイプ、ステータスなど、設定されたビューに関する情報を表示します。

SNMPv3 の設定：例

エンジン ID の設定

次に、ローカル SNMP エンジンの ID を設定する例を示します。

```
snmp-server engineID local 00:00:00:09:00:00:00:a1:61:6c:20:61
```

（注） エンジン ID が設定されると、SNMP エージェントが再起動します。

ローカル SNMP エンジンの ID の確認

次に、ローカル SNMP エンジンの ID を確認する例を示します。

```
config
  show snmp engineid

  SNMP engineID 00000009000000a1ffffffff
```

ビューの作成

ビューを作成するには 2 つの方法があります。

- **snmp-server view** コマンドの **included** キーワードを使用することによって、ビューに MIB ファミリの ASN.1 サブツリーのオブジェクト識別子（OID）を包含することができます。
- **snmp-server view** コマンドの **excluded** キーワードを使用することによって、ビューから MIB ファミリの ASN.1 サブツリーの OID サブツリーを除外することができます。

次に、sysName（1.3.6.1.2.1.1.5）オブジェクトを含むビューを作成する例を示します。

```
config
  snmp-server view SNMP_VIEW1 1.3.6.1.2.1.1.5 included
```

次に、システム グループのすべての OID を含むビューを作成する例を示します。

```
config
  snmp-server view SNMP_VIEW1 1.3.6.1.2.1.1 included
```

次に、除外されている sysName オブジェクト（1.3.6.1.2.1.1.5）を除く、システム グループのすべての OID を含むビューを作成する例を示します。

```
config
  snmp-server view SNMP_VIEW1 1.3.6.1.2.1.1 included
  snmp-server view SNMP_VIEW1 1.3.6.1.2.1.1.5 excluded
```

設定したビューの確認

次に、設定したビューの情報を表示する例を示します。

```
RP/0/RP0/CPU0:router# show snmp view

  v1default 1.3.6.1 - included nonVolatile active
  SNMP_VIEW1 1.3.6.1.2.1.1 - included nonVolatile active
  SNMP_VIEW1 1.3.6.1.2.1.1.5 - excluded nonVolatile active
```

グループの作成

通知、読み取り、または書き込みビューを明示的に指定しないと、Cisco IOS XR ソフトウェアではv1デフォルト（1.3.6.1）が使用されます。次に、デフォルトビューを使用するグループを作成する例を示します。

```
RP/0/RP0/CPU0:router# snmp-server group group-name v3 auth
```

次の設定例は、グループに適用されるビューから除外された sysUpTime オブジェクト（1.3.6.1.2.1.1.3）を除く、システム内のすべての OID に対する読み取りアクセス権があり、sysName オブジェクト（1.3.6.1.2.1.1.5）に対しては書き込みアクセス権しかないグループを作成する例を示します。

```
!
  snmp-server view view_name1 1.3.6.1.2.1.1 included
  snmp-server view view_name1 1.3.6.1.2.1.1.3 excluded
  snmp-server view view_name2 1.3.6.1.2.1.1.5 included
  snmp-server group group_name1 v3 auth read view_name1 write view_name2
  !
```

グループの確認

この例では、設定したグループの属性を確認する方法を示します。

```
RP/0/RP0/CPU0:router# show snmp group

  groupname: group_name1                  security model:usm
  readview : view_name1                   writeview: view_name2
  notifyview: v1default
  row status: nonVolatile
```

ユーザの作成および確認

次の SNMPv3 ビューおよび SNMPv3 グループの設定があるとします。

```
!
  snmp-server view view_name 1.3.6.1.2.1.1 included
  snmp-server group group_name v3 noauth read view_name write view-name
  !
```

次に、システム グループに対する読み取りビュー アクセスおよび書き込みビュー アクセスの権限を持つ noAuthNoPriv ユーザを作成する例を示します。

```
config
  snmp-server user noauthuser group_name v3
```

（注） noAuthNoPriv ユーザを作成するには、ユーザが noauth グループに属している必要があります。

次に、SNMP ユーザに適用する属性を確認する例を示します。

```
RP/0/RP0/CPU0:router# show snmp user

  User name: noauthuser
  Engine ID: localSnmpID
  storage-type: nonvolatile active
```

次の SNMPv3 ビューおよび SNMPv3 グループの設定があるとします。

```
!
  snmp-server view SNMP_VIEW1 1.3.6.1.2.1.1 included
  snmp-server group SNMP_GROUP1 v3 auth notify SNMP_VIEW1 read SNMP_VIEW1 write SNMP_VIEW1

 !
```

次に、システム グループに対する認証（暗号化を含む）、読み取り/書き込みビュー アクセスの権限を持つユーザを作成する例を示します。

```
config
  snmp-server user userv3authpriv SNMP_GROUP1 v3 auth md5 password123 priv aes 128
password123
```

次の SNMPv3 ビューおよび SNMPv3 グループの設定があるとします。

```
!
  snmp-server view view_name 1.3.6.1.2.1.1 included
  snmp group group_name v3 priv read view_name write view_name
  !
```

次に、システム グループに対する読み取りビュー アクセスおよび書き込みビュー アクセスの権限を持つ authNoPriv ユーザを作成する例を示します。

```
RP/0/RP0/CPU0:router# snmp-server user authuser group_name v3 auth md5 clear auth_passwd
```

（注） グループはセキュリティ レベル Auth に設定されているので、このグループにアクセスするには、ユーザが最低でも「auth」として設定されている必要があります（「priv」ユーザもこのグループにアクセスできます）。このグループに設定された authNoPriv ユーザの authuser は、ビューにアクセスするために認証パスワードを入力する必要があります。この例では、auth_passwd が認証パスワード文字列として設定されています。auth_passwd パスワード文字列の前に **clear** キーワードが指定されていることに注意してください。**clear** キーワードは、入力されているパスワード文字列が暗号化されていないことを示しています。

次に、SNMP ユーザに適用する属性を確認する例を示します。

```
RP/0/RP0/CPU0:router# show snmp user

  User name: authuser
  Engine ID: localSnmpID
  storage-type: nonvolatile active
```

次の SNMPv3 ビューおよび SNMPv3 グループの設定があるとします。

```
!
  snmp view view_name 1.3.6.1.2.1.1 included
  snmp group group_name v3 priv read view_name write view_name
  !
```

次に、システム グループへの読み取りビュー アクセスおよび書き込みビュー アクセスの権限を持つ authPriv ユーザを作成する例を示します。

```
config
  snmp-server user privuser group_name v3 auth md5 clear auth_passwd priv des56 clear
priv_passwd
```

（注） グループのセキュリティ レベルは Priv なので、ユーザがこのグループにアクセスするには、「priv」ユーザとして設定される必要があります。この例のユーザ privuser は、ビュー内の OID にアクセスするために、認証パスワードとプライバシー パスワードの両方を入力する必要があります。

次に、SNMP ユーザに適用する属性を確認する例を示します。

```
RP/0/RP0/CPU0:router# show snmp user

  User name: privuser
  Engine ID: localSnmpID
  storage-type: nonvolatile active
```

SNMP トラップ通知の設定

ここでは、SNMP トラップ通知を送信するようにルータを設定する方法について説明します。

（注） SNMPv3 の設定 （10 ページ） タスクで説明した手順をすでに完了している場合は、SNMPv3 の設定 （10 ページ） を省略できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**snmp-servergroup***name*{**v1v2v3**{**auth** | **noauth** | **priv**}}[**read***view*]**write***view*] [**notify***view*] [*access-list-name*] 例： `RP/0/RP0/CPU0:router# snmp-server group group_name v3 noauth read view_name1 writer view_name2`	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。
ステップ 3	**snmp-serveruser***groupname*{**v1v2cv3**{**auth** | **md5** | **sha**}{**clear** | **encrypted**}*auth-password*] [**privdes56** {**clear** | *access-list-name*] 例： `RP/0/RP0/CPU0:router# snmp-server group group_name v3 noauth read view_name1 writer view_name2`	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。
ステップ 4	**snmp-serveruser***usernamegroupname*{**v1v2cv3**{**auth** | **md5** | **sha**}{**clear** | **encrypted**}*auth-password*] [**privdes56** {**clear** | *access-list-name*] 例： `RP/0/RP0/CPU0:routerconfig# snmp-server user noauthuser group_name v3`	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。
ステップ 5	[**snmp-serverhost** *address* [**traps**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*] 例： `RP/0/RP0/CPU0:router(config)# snmp-server host 12.26.25.61 traps version 3 noauth userV3noauth`	SNMP トラップ通知、使用する SNMP のバージョン、通知のセキュリティ レベル、通知の受信者（ホスト）を指定します。

	コマンドまたはアクション	目的
ステップ 6	**snmp-servertraps** [*notification-type*] 例： `RP/0/RP0/CPU0:router(config)# snmp-server traps bgp`	トラップ通知の送信をイネーブルにし、送信するトラップ通知のタイプを指定します。 • トラップを *notification-type* 引数で指定しない場合は、サポートされるすべてのトラップ通知がルータ上でイネーブルになります。ルータで使用可能なトラップ通知を表示するには、**snmp-servertraps?** コマンドを入力します。
ステップ 7	**commit**	
ステップ 8	（任意） **showsnmphost** 例： `RP/0/RP0/CPU0:router# show snmp host`	設定された SNMP 通知の受信者（ホスト）、ポート番号、セキュリティ モデルに関する情報を表示します。

トラップ通知の設定：例

次に、異なるタイプのトラップを送信するように SNMP エージェントを設定する例を示します。設定には、v2c ユーザ、noAuthNoPriv ユーザ、anauthNoPriv ユーザ、および AuthPriv ユーザが含まれます。

（注） デフォルトのユーザ データグラム プロトコル（UDP）ポートは 161 です。**udp-port** キーワードおよび *port* 引数を指定して UDP ポートを指定しないと、設定された SNMP トラップ通知はポート 161 に送信されます。

```
!
  snmp-server host 10.50.32.170 version 2c userv2c udp-port 2345
  snmp-server host 10.50.32.170 version 3 auth userV3auth udp-port 2345
  snmp-server host 10.50.32.170 version 3 priv userV3priv udp-port 2345
  snmp-server host 10.50.32.170 version 3 noauth userV3noauth udp-port 2345
  snmp-server user userv2c groupv2c v2c
  snmp-server user userV3auth groupV3auth v3 auth md5 encrypted 140F0A13
  snmp-server user userV3priv groupV3priv v3 auth md5 encrypted 021E1C43 priv des56
encrypted 1110001C
  snmp-server user userV3noauth groupV3noauth v3 LROwner
  snmp-server view view_name 1.3 included
  snmp-server community public RW
  snmp-server group groupv2c v2c read view_name
  snmp-server group groupV3auth v3 auth read view_name
  snmp-server group groupV3priv v3 priv read view_name
  snmp-server group groupV3noauth v3 noauth read view_name
  !
```

次に、SNMP トラップ通知の受信者ホストの設定、つまり SNMP トラップ通知の受信者を確認する方法を示しています。出力には、次の情報が表示されます。

- 設定された通知ホストの IP アドレス
- SNMP 通知メッセージが送信される UDP ポート
- 設定されたトラップのタイプ
- 設定されたユーザのセキュリティ レベル
- 設定されたセキュリティ モデル

```
config
  show snmp host

  Notification host: 10.50.32.170 udp-port: 2345 type: trap
  user: userV3auth security model: v3 auth

  Notification host: 10.50.32.170 udp-port: 2345 type: trap
  user: userV3noauth security model: v3 noauth

  Notification host: 10.50.32.170 udp-port: 2345 type: trap
  user: userV3priv security model: v3 priv

  Notification host: 10.50.32.170 udp-port: 2345 type: trap
  user: userv2c security model: v2c
```

SNMP エージェントの連絡先、場所、およびシリアル番号の設定

このタスクは、SNMPエージェントのシステムの連絡先文字列、システムの場所の文字列、およびシステム シリアル番号を設定する方法について説明します。

（注） ここで **snmp-server** コマンドを実行する順序は重要ではありません。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	（任意） **snmp-servercontact** *system-contact-string* **例**： `RP/0/RP0/CPU0:router(config)# snmp-server contact Dial System Operator at beeper # 27345`	システムの連絡先文字列を設定します。
ステップ 3	（任意） **snmp-serverlocation** *system-location* **例**：	システムの場所を表す文字列を設定します。

	コマンドまたはアクション	目的
	`RP/0/RP0/CPU0:router(config)# snmp-server location Building 3/Room 214`	
ステップ 4	（任意） **snmp-serverchassis-id** *serial-number* 例： `RP/0/RP0/CPU0:router(config)# snmp-server chassis-id 1234456`	システムのシリアル番号を設定します。
ステップ 5	**commit**	

SNMP エージェント パケットの最大サイズの定義

このタスクでは、SNMP サーバが要求を受信しているか応答を生成しているときに、許可される SNMP パケットの最大サイズを設定する例を示します。

（注） ここで **snmp-server** コマンドを実行する順序は重要ではありません。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	（任意） **snmp-serverpacketsize** *byte-count* 例： `RP/0/RP0/CPU0:router(config)# snmp-server packetsize 1024`	最大パケット サイズを設定します。
ステップ 3	**commit**	

通知操作値の変更

SNMP 通知がイネーブルになると、送信元インターフェイス、メッセージキューの長さ、または再送信間隔にデフォルト以外の値を指定することができます。

ここでは、トラップ通知用の送信元インターフェイス、各ホストのメッセージキューの長さ、および再送信間隔を指定する方法について説明します。

（注） ここで **snmp-server** コマンドを実行する順序は重要ではありません。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	（任意） **snmp-servertrap-source** *type interface-path-id* 例： `RP/0/RP0/CPU0:router(config)# snmp-server trap-source POS 0/0/1/0`	トラップ通知の送信元インターフェイスを指定します。
ステップ 3	（任意） **snmp-serverqueue-length** *length* 例： `RP/0/RP0/CPU0:router(config)# snmp-server queue-length 20`	各通知のメッセージ キューの長さを設定します。
ステップ 4	（任意） **snmp-servertrap-timeout** *seconds* 例： `RP/0/RP0/CPU0:router(config)# snmp-server trap-timeout 20`	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 5	**commit**	

IP precedence および DSCP 値の設定

ここでは、SNMP トラフィックに対して IP precedence または IP DSCP を設定する方法について説明します。

始める前に

SNMP が設定されていること。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	

	コマンドまたはアクション	目的
ステップ 2	次のいずれかのコマンドを使用します。 • **snmp-serveripv4precedence** *value* • **snmp-serveripv4dscp** *value* 例： `RP/0/RP0/CPU0:router(config)# snmp-server dscp 24`	SNMP トラフィックの IP precedence または IP DSCP 値を設定します。
ステップ 3	**commit**	

SNMP トラフィックの IP precedence 値の設定：例

次の例に、SNMP IP precedence 値を 7 に設定する方法を示します。

```
configure
  snmp-server ipv4 precedence 7
  exit

  Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]: y
```

SNMP トラフィックの IP DSCP 値の設定：例

次の例に、SNMP トラフィックの IP DSCP 値を 45 に設定する方法を示します。

```
configure
  snmp-server ipv4 dscp 45
  exit

  Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]: y
```

SNMP コンテキスト マッピングの表示

SNMPエージェントは、クライアント機能により作成されたSNMPコンテキストに基づいてクエリーを提供します。コンテキスト マッピング テーブルが存在します。コンテキスト マッピングテーブルの各エントリには、コンテキスト名、コンテキストを作成した機能の名前、および機能の特定のインスタンスの名前が含まれます。

手順

	コマンドまたはアクション	目的
ステップ 1	**showsnmpcontext-mapping** 例： `RP/0/RP0/CPU0:router# show snmp context-mapping`	SNMP コンテキスト マッピング テーブルを表示します。

パケット損失のモニタリング

パケット損失が指定したしきい値を超えたときの SNMP トラップの生成を設定することにより、パケット損失をモニタすることが可能です。このタスクで説明する設定は、EVENT-MIB の MIB テーブルのエントリの作成をイネーブルにします。これは、その後 SNMP GET 操作を使用してパケット損失をモニタできます。

始める前に

（注） このタスクで説明する設定を使用して、EVENT-MIB MIB テーブルに作成されたエントリは、SNMP SET を使用して変更できません。

SNMP SET を使用して作成された、EVENT-MIB MIB テーブルへのエントリは、このタスクで説明する設定を使用して変更できません。

手順

	コマンドまたはアクション	目的
ステップ 1	**snmp-server mibs eventmib packet-loss** *type interface-path-id***falling** *lower-threshold***interval** *sampling-interval***rising** *upper-threshold* 例： `RP/0/RP0/CPU0:router(config)# snmp-server mibs eventmib packet-loss falling 1 interval 5 rising 2`	パケット損失が指定したしきい値を超えたときに、インターフェイスに対して SNMP EVENT-MIB トラップを生成します。最大 100 のインターフェイスをモニタできます。 **falling** *lower-threshold*：低い方のしきい値を指定します。2 つの間隔間のパケット損失がこのしきい値を下回り、以前に mteTriggerRising トラップが生成されていた場合、SNMP の mteTriggerFalling トラップが生成されます。このトラップは、パケット損失が高い方のしきい値を超えて、その後、低い方のしきい値を再度下回るまで生成されません。

	コマンドまたはアクション	目的
		interval *sampling-interval*：パケット損失の統計情報がポーリングされる頻度を指定します。これは、5 ～ 1440 分の 5 の倍数の値です。 **rising** *upper-threshold*：高い方のしきい値を指定します。2 つの間隔間のパケット損失がこのしきい値を超えると、SNMP の mteTriggreRising トラップが生成されます。このトラップは、パケット損失が下限しきい値を下回ってから、上限しきい値を上回るまで生成されません。

維持する MIB データの設定

SNMP MIB 定義では、多くの場合、オブジェクト テーブルに任意の 32 ビットのインデックスを定義しています。MIB の実装では、多くの場合、MIB インデックスから内部データ構造へのマッピングを行います。このデータ構造は他のデータ セットのキーになります。このような MIB テーブルでは、テーブル内に含まれるデータが、モデル化されている他の要素の識別子となっている場合があります。たとえば、ENTITY-MIB においては、entPhysicalTable のエントリは 31 ビットの値である entPhysicalIndex によってインデックス化されていますが、このエントリは entPhysicalName またはテーブル内の他のオブジェクトの組み合わせによって識別することができます。

一部の MIB テーブルのサイズが原因で、32 ビット MIB インデックスから、ネットワーク管理ステーションがエントリを識別できる他のデータへのすべてのマッピングを検出するには、膨大な処理が必要になります。そのため、プロセスの再開、リスタート、スイッチオーバー、デバイスのリロードを行っても、一部の MIB インデックスが維持される必要が生じます。ENTITY-MIB の entPhysicalTable および CISCO-CLASS-BASED-QOS-MIB は、このような MIB の例であり、インデックス値を維持する必要が生じる場合が多くあります。

また、CISCO-CLASS-BASED-QOS-MIB 統計情報のクエリ実行時のクエリの応答時間や CPU 使用率の問題により、サービス ポリシーの統計情報はキャッシュしておくことが望ましいと言えます。

手順

	コマンドまたはアクション	目的
ステップ 1	（任意） **snmp-server entityindex persist** **例**： `RP/0/RP0/CPU0:router(config)# snmp-server entityindex persist`	ENTITY-MIB データの固定ストレージ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	（任意） **snmp-server mibs cbqosmib persist** 例： `RP/0/RP0/CPU0:router(config)# snmp-server mibs cbqosmib persist`	CISCO-CLASS-BASED-QOS-MIB データの固定ストレージをイネーブルにします。
ステップ 3	（任意） **snmp-server cbqosmib cache refresh time** *time* 例： `RP/0/RP0/CPU0:router(config)# snmp-server mibs cbqosmib cache refresh time 45`	QoS MIB のキャッシュをイネーブルにして、キャッシュのリフレッシュ時間を設定します。
ステップ 4	（任意） **snmp-server cbqosmib cache service-policy count** *count* 例： `RP/0/RP0/CPU0:router(config)# snmp-server mibs cbqosmib cache service-policy count 50`	QoS MIB のキャッシュをイネーブルにして、キャッシュするサービスポリシーの数に制限を設けます。
ステップ 5	**snmp-server ifindex persist** 例： `RP/0/RP0/CPU0:router(config)# snmp-server ifindex persist`	すべての簡易ネットワーク管理プロトコル（SNMP）インターフェイスで、ifIndex パーシステンスをグローバルにイネーブルにします。

インターフェイスのサブセットに対する linkUp および linkDown トラップの設定

トラップを設定するインターフェイスを表すための正規表現を指定することで、同時に多数のインターフェイスに対して linkUp および linkDown トラップをイネーブルまたはディセーブルにすることができます。

始める前に

SNMP が設定されていること。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	

	コマンドまたはアクション	目的
ステップ 2	**snmp-server interface subset** *subset-number***regular-expression** *expression* 例： `RP/0/RP0/CPU0:router(config)# snmp-server interface subset 10 regular-expression "^Gig[a-zA-Z]+[0-9/]+\." RP/0/RP0/CPU0:router(config-snmp-if-subset)#`	正規表現で識別されたインターフェイスに対し、snmp-server インターフェイスモードを開始します。 subset-number 引数は、インターフェイスのセットを識別し、インターフェイスが複数のサブセットに含まれている場合は、そのサブセットのプライオリティも割り当てます。数値が小さいほどプライオリティが高く、そのコンフィギュレーションは数値が大きいインターフェイスサブセットよりも優先されます。 *expression* 引数は二重引用符で囲んで入力する必要があります。 正規表現の詳細については、の「*Understanding Regular Expressions, Special Characters, and Patterns*」モジュールを参照してください。
ステップ 3	**notification linkupdown disable** 例： `RP/0/RP0/CPU0:router(config-snmp-if-subset)# notification linkupdown disable`	設定しているすべてのインターフェイスに対して linkUp および linkDown トラップをディセーブルにします。ディセーブルにしたインターフェイスをイネーブルにするには、このコマンドの **no** 形式を使用します。
ステップ 4	**commit**	
ステップ 5	（任意） **show snmp interface notification subset** *subset-number* 例： `RP/0/RP0/CPU0:router# show snmp interface notification subset 10`	サブセットのプライオリティで識別されたすべてのインターフェイスについて、linkUp および linkDown 通知のステータスを表示します。
ステップ 6	（任意） **show snmp interface notification regular-expression** *expression* 例： `RP/0/RP0/CPU0:router# show snmp interface notification regular-expression "^Gig[a-zA-Z]+[0-9/]+\."`	正規表現で識別されたすべてのインターフェイスについて、linkUp および linkDown 通知のステータスを表示します。

	コマンドまたはアクション	目的
ステップ 7	（任意） **show snmp interface notification** *type interface-path-id* 例： `RP/0/RP0/CPU0:router# show snmp interface notification tengige 0/0/0/0.10`	指定されたインターフェイスについて、linkUp および linkDown 通知のステータスを表示します。