SNMP の設定

この章では、Cisco NX-OS デバイス上で SNMP 機能を設定する方法について説明します。

この章は、次の内容で構成されています。

SNMP について

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 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。

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 を使用してネットワークデバイスのアクティビティを制御し、モニタリングするシステム
- SNMP エージェント：デバイスのデータを維持し、必要に応じてこれらのデータを管理システムに報告する、管理対象デバイス内のソフトウェア コンポーネント。Cisco Nexus デバイスはエージェントおよび MIB をサポートします。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（Management Information Base; 管理情報ベース）：SNMP エージェントの管理対象オブジェクトの集まり

SNMP は、RFC 3411 ～ 3418 で規定されています。

デバイスは、SNMPv1、SNMPv2c、およびSNMPv3をサポートします。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。

Cisco NX-OS は IPv6 による SNMP をサポートしています。

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。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。

Cisco NX-OS は、トラップまたはインフォームとして SNMP 通知を生成します。トラップは、エージェントからホスト レシーバ テーブルで指定された SNMP マネージャに送信される、非同期の非確認応答メッセージです。インフォームは、SNMP エージェントから SNMP マネージャに送信される非同期メッセージで、マネージャは受信したという確認応答が必要です。

トラップの信頼性はインフォームより低くなります。SNMPマネージャはトラップを受信しても確認応答（ACK）を送信しないからです。デバイスは、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。デバイス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。

複数のホスト レシーバーに通知を送信するよう Cisco NX-OS を設定できます。

次の表は、デフォルトで有効になっている SNMP トラップを示します。

Trap Type	説明
全体	: coldStart
エンティティ	: entity_fan_status_change
エンティティ	: entity_mib_change
エンティティ	: entity_module_status_change
エンティティ	: entity_module_inserted
エンティティ	: entity_module_removed
エンティティ	: entity_power_out_change
エンティティ	: entity_power_status_change
エンティティ	: entity_unrecognised_module
リンク	: cErrDisableInterfaceEventRev1

Trap Type	説明
リンク	: cieLinkDown
リンク	: cieLinkUp
リンク	: cmn-mac-move-notification
リンク	: delayed-link-state-change
リンク	: extended-linkDown
リンク	: extended-linkUp
リンク	: linkDown
リンク	: linkUp
rf	: redundancy_framework
upgrade	: UpgradeOpNotifyOnCompletion
upgrade	: UpgradeJobStatusNotify
エンティティ	: entity_sensor
rmon	: fallingAlarm
rmon	: hcRisingAlarm
rmon	: hcFallingAlarm
rmon	: risingAlarm

SNMPv3

SNMPv3は、ネットワーク経由のフレームの認証と暗号化を組み合わせることによって、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：メッセージのソースが有効かどうかを判別します。
- 暗号化：許可されていないソースにより判読されないように、パケットの内容のスクランブルを行います。

SNMPv3では、セキュリティモデルとセキュリティレベルの両方が提供されています。セキュリティモデルは、ユーザおよびユーザが属するロールを設定する認証方式です。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、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。このレベルは、SNMPv3 ではサポートされていません。
- 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。次の表に、セキュリティ モデルとレベルの組み合わせの意味を示します。

***Table 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	コミュニティ ストリング	なし	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	コミュニティ ストリング	なし	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	authNoPriv	HMAC-MD5、HMAC-SHA、または SHA-256	未対応	Hash-Based Message Authentication Code（HMAC）メッセージ ダイジェスト 5（MD5）アルゴリズムまたは HMAC Secure Hash Algorithm（SHA）アルゴリズムに基づいて認証します。

モデル	レベル	認証	暗号化	結果
v3	authPriv	HMAC-MD5、HMAC-SHA、または SHA-256	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。データ暗号規格（DES）の 56 ビット暗号化、および暗号ブロック連鎖（CBC）DES（DES-56）標準に基づいた認証を提供します。

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。

Cisco NX-OS は、SNMPv3 に 3 つの認証プロトコルを使用します。

- HMAC-MD5-96 認証プロトコル
- HMAC-SHA-96 認証プロトコル
- SHA-256 認証プロトコル

Cisco NX-OS リリース 9.3(7) 以降では、SNMPv3 に HMAC-SHA-256 認証プロトコルが使用されます。

Note SHA-256 SNMP ユーザがスイッチで設定されている場合、ISSD は **install all** コマンドを使用することを推奨します。そうしないと、設定が失われます。

Cisco NX-OS は、SNMPv3 メッセージ暗号化用プライバシー プロトコルの 1 つとして、Advanced Encryption Standard（AES）を使用し、RFC 3826 に準拠します。

priv オプションで、SNMP セキュリティ暗号化方式として、DES または 128 ビット AES 暗号化を選択できます。**priv** オプションおよび **aes-128** トークンは、128 ビットの AES キーを生成するためのプライバシ パスワードであることを示します。AES のプライバシー パスワードは最小で 8 文字です。パスフレーズをクリアテキストで指定する場合は、大文字と小文字を区別して、最大 64 文字の英数字を指定できます。ローカライズドキーを使用する場合は、最大 130 文字を指定できます。

Note 外部の AAA サーバを使用して SNMPv3 を使う場合、外部 AAA サーバのユーザー設定でプライバシー プロトコルに AES を指定する必要があります。

CLI および SNMP ユーザの同期

SNMPv3 ユーザ管理は、Access Authentication and Accounting（AAA）サーバ レベルで集中化できます。この中央集中型ユーザ管理により、Cisco NX-OSの SNMP エージェントは AAA サーバのユーザ認証サービスを利用できます。ユーザ認証が検証されると、SNMP PDU の処理が進行します。AAA サーバはユーザ グループ名の格納にも使用されます。SNMP はグループ名を使用して、スイッチでローカルに使用できるアクセス ポリシーまたはロール ポリシーを適用します。

ユーザ グループ、ロール、またはパスワードの設定が変更されると、SNMP と AAA の両方のデータベースが同期化されます。

Cisco NX-OS は、次のようにユーザ設定を同期化します。

- **snmp-server user** コマンドで指定された認証パスフレーズは、CLI ユーザのパスワードになります。
- **username** コマンドで指定されたパスワードが SNMP ユーザの認証およびプライバシー パスフレーズになります。
- SNMP または CLI を使用してユーザを作成または削除すると、SNMP と CLI の両方でユーザが作成または削除されます。
- ユーザとロールの対応関係の変更は、SNMP と CLI で同期化されます。
- CLI から行ったロール変更（削除または変更）は、SNMP と同期します。

Note パスフレーズまたはパスワードをローカライズしたキーおよび暗号形式で設定した場合、Cisco NX-OS はユーザ情報（パスワードやロールなど）を同期させません。

Cisco NX-OS はデフォルトで、同期したユーザ設定を 60 分間維持します。

セキュリティおよび SNMP ユーザーの同期の無効化

Cisco NX-OS リリース 10.2(2)F 以降、SNMP とセキュリティ（AAA または CLI）コンポーネント間のユーザー同期を無効にするオプションを提供するために、次の同期解除コマンドが導入されました。

snmp-server disable snmp-aaa sync

このコマンドは、Nexus スイッチの構成端末から実行できます。デフォルトでは、desynchronization コマンドの **no** 形式がスイッチで使用できます。

デバイスで同期解除コマンドの no 形式が有効になっている場合、たとえば `switch (config)#` **no snmp-server disable snmp-aaa sync** の場合には、実行構成におけるそのユーザーの **username** 作成で、 **snmp-server user** CLI の結果を利用してユーザーを作成することができます。また、逆も可能です。したがって、ユーザーは、作成/更新時に **snmp-server user** CLI または **username** CLI に記載されている認証資格情報を使用してスイッチにログインできます。スイッチのネットワーク マネージャから SNMP 操作を実行することもできます。したがって、desynchronization コマンドの **no** 形式を使用すると、SNMP と AAA 間のユーザー同期は、10.2(2)F より前のリリースと同じように機能します。

デバイスで同期解除コマンドが有効になっている場合、たとえば `switch (config)#` **snmp-server disable snmp-aaa sync** の場合には、**snmp-server user** コマンドによって作成されたユーザーに対し、ユーザー名構成は作成されません。したがって、ユーザーはスイッチにログインできず、スイッチのネットワーク マネージャを介して SNMP 操作を実行することのみが許可されます。同様に、**username** CLI を使用してセキュリティ ユーザーを作成しても、そのユーザーに対応する **snmp-server user** CLI は作成されません。このユーザーはスイッチにログインできますが、スイッチで SNMP 操作を実行することはできません。これは、desynchronization コマンドによってリリース 10.2(2)F から導入された新機能です。

非同期コマンドのステータスは、次のいずれかの方法で確認できます：

- CLI **show snmp internal globals** の出力にある `SNMP-AAA sync disable` フィールドの値
- `sys/snmp/inst/globals` MO のフィールド `disableSnmpAaaSync` の値
- コマンドが有効か無効かに応じて、CLI は **show-running-config** 出力および **show-running-config-snmp** 出力または **show-running-all** 出力にそれぞれ出力します。

リモートユーザー

RADIUS や TACACS+ などのプロトコルを使用して外部サーバー経由でログイン認証されているリモートユーザーに関しては、スイッチで同期解除コマンドが有効になっている場合、SNMP でリモート ユーザーを作成できません。詳細については、*Cisco Nexus 9000 NX-OS Security Configuration Guide* の *AAA* を構成するの章を参照してください。

ただし、スイッチで desynchronization コマンドの **no** 形式が有効になっている場合、リモートユーザが AAA で作成されると、対応するユーザが SNMP でも作成されます。さらに、ユーザーは SNMP の running-config 出力には表示されませんが、管理対象デバイスで SNMP 操作を実行できます。これは、リリース 10.2(2)F より前からの既存の機能です。

DCNM セキュリティ ユーザー

desynchronization コマンドが有効になっている場合、DCNM（リリース 12.0.1a 以降は Nexus Dashboard Fabric Controller とも呼ばれます）を使用して作成されたセキュリティ ユーザーには、対応する SNMPv3 プロファイルがありません。同期が無効になっている場合、セキュリティ コンポーネントで作成されたユーザーはスイッチにログインできますが、コントローラはスイッチを検出しません。コントローラは、セキュリティ ユーザー用に作成された SNMP 構成を使用してスイッチを検出するためです。さらに、SNMP は、userDB の非同期状態のため、作成されたセキュリティ ユーザーを認識しないので、スイッチを検出できません。したがって、コントローラによってスイッチが検出されるようにするには、SNMP ユーザーを明示的に作成する必要があります。DCNM 機能とともに desynchronization コマンドを使用することはお勧めしません。詳細については、*Cisco Nexus 9000 NX-OS Security Configuration Guide* を参照してください。

ISSD と ISSU

一般に、SNMP ユーザーの同期が無効になっている場合は、非同期のユーザーをすべて削除しない限り、SNMP ユーザーの同期を有効にしないでください。このような組み合わせの実行コンフィギュレーションでは、設定の置換が失敗します。

古いリリースで、同期解除コマンドを使用せずに同期解除状態を実現する唯一の方法は、次のとおりです。

- 同期解除状態のリリースから、同期解除コマンドの存在しないリリースへ、中断を伴う/伴わない ISSD を実行します。同期解除されたデータベースは、ISSD により以前のリリースにそのまま取り込まれます。

（注） そのような ISSD の後にユーザー データベースに加えられた変更は、SNMP とセキュリティ コンポーネントの間で同期されます。

このような ISSD の後、同期解除コマンドの存在するリリースへの ISSU を実行すると、同期解除されたユーザー データベースがそのまま取り込まれます。一方、同期解除コマンドはデフォルトの **no** 形式で起動します。必要に応じて、同期解除コマンドを有効にしてください。

グループベースの SNMP アクセス

Note グループが業界全体で使用されている標準 SNMP 用語なので、この SNMP の項では、ロールのことをグループと言います。

SNMP アクセス権は、グループ別に編成されます。SNMP 内の各グループは、CLI を使用する場合のロールに似ています。各グループは読み取りアクセス権または読み取りと書き込みアクセス権を指定して定義します。

ユーザ名が作成され、ユーザのロールが管理者によって設定され、ユーザがそのロールに追加されていれば、そのユーザはエージェントとの通信を開始できます。

SNMP および Embedded Event Manager

Embedded Event Manager（EEM）機能は、SNMP MIB オブジェクトを含むイベントをモニタし、これらのイベントに基づいてアクションを開始します。SNMP 通知の送信もアクションの 1 つです。EEM は SNMP 通知として、CISCO-EMBEDDED-EVENT-MGR-MIB の cEventMgrPolicyEvent を送信します。

マルチ インスタンス サポート

デバイスは、プロトコルインスタンスや仮想ルーティングおよびフォワーディング（VRF）インスタンスなどの論理ネットワーク エンティティの複数のインスタンスをサポートできます。大部分の既存 MIB は、これら複数の論理ネットワーク エンティティを識別できません。たとえば、元々の OSPF-MIB ではデバイス上のプロトコル インスタンスが 1 つであることが前提になりますが、現在はデバイス上で複数の OSPF インスタンスを設定できます。

SNMPv3 ではコンテキストを使用して、複数のインスタンスを識別します。SNMP コンテキストは管理情報のコレクションであり、SNMP エージェントを通じてアクセスできます。デバイスは、さまざまな論理ネットワーク エンティティの複数のコンテキストをサポートできます。SNMP コンテキストによって、SNMP マネージャはさまざまな論理ネットワーク エンティティに対応するデバイス上でサポートされる、MIB モジュールの複数のインスタンスの 1 つにアクセスできます。

Cisco NX-OS は、SNMP コンテキストと論理ネットワーク エンティティ間のマッピングのために、CISCO-CONTEXT-MAPPING-MIB をサポートします。SNMP コンテキストは VRF、プロトコル インスタンス、またはトポロジに関連付けることができます。

SNMPv3 は、SNMPv3 PDU の contextName フィールドでコンテキストをサポートします。この contextName フィールドを特定のプロトコルインスタンスまたは VRF にマッピングできます。

SNMPv2c の場合は、SNMP-COMMUNITY-MIB の snmpCommunityContextName MIB オブジェクトを使用して、SNMP コミュニティをコンテキストにマッピングできます（RFC 3584）。さらに CISCO-CONTEXT-MAPPING-MIB または CLI を使用すると、この snmpCommunityContextName を特定のプロトコル インスタンスまたは VRF にマッピングできます。

SNMP のハイ アベイラビリティ

Cisco NX-OS は、SNMP のステートレス リスタートをサポートします。リブートまたはスーパーバイザ スイッチオーバーの後、Cisco NX-OS は実行コンフィギュレーションを適用します。

SNMP の仮想化サポート

Cisco NX-OS は、SNMP のインスタンスを 1 つサポートします。SNMP は複数の MIB モジュールインスタンスをサポートし、それらを論理ネットワーク エンティティにマッピングします。

SNMP も VRF を認識します。特定の VRF を使用して、SNMP 通知ホスト レシーバに接続するように SNMP を設定できます。通知が発生した VRF に基づいて、SNMP ホスト レシーバへの通知をフィルタリングするように SNMP を設定することもできます。

SNMP の注意事項および制約事項

SNMP には、次の注意事項および制限事項があります。

- SNMP SET を使用して構成されたコマンドは、SNMP SET のみを使用して削除する必要があります。コマンド ライン インターフェイス（CLI）または NX-API を使用して構成されたコマンドは、CLI または NX-API のみを使用して削除する必要があります。
- クリア テキスト パスワードを使用して AAA でユーザーを作成または編集すると、SNMP はデフォルトの認証（md5）および priv タイプを持つユーザーを作成または編集します。

  クリア テキスト パスワードを使用して SNMP でユーザーを作成または編集すると、AAA はデフォルトのパスワード タイプ（タイプ 5）を持つユーザーを作成または編集します。
- アクセス コントロール リスト（ACL）は、スイッチに設定されたローカル SNMPv3 ユーザのみに適用できます。ACL は、認証、許可、アカウンティング（AAA）サーバに保存されるリモート SNMPv3 ユーザに適用できません。
- 同期解除されたすべてのユーザが削除されない限り、SNMP ユーザ同期を無効にした後は有効にしないでください。このような組み合わせの実行コンフィギュレーションでは、設定の置換が失敗します。
- Cisco NX-OS は、一部の SNMP MIB への読み取り専用アクセスをサポートします。詳細については次の URL にアクセスして、Cisco NX-OS の MIB サポート リストを参照してください。https://cisco.github.io/cisco-mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html
- Cisco NX-OS は、SNMPv3 noAuthNoPriv セキュリティ レベルをサポートしていません。
- Cisco Nexus 9000 シリーズ スイッチと、Cisco Nexus 3164Q、31128PQ、3232C、3264Q スイッチは、SNMP ローカル エンジンID の設定をサポートしています。
- 以前のリリースへの無停止ダウングレード パスを行う場合、ローカル エンジン ID を設定していたなら、ローカル エンジン ID の設定を戻してから、SNMP ユーザとコミュニティ文字列を再設定する必要があります。
- 特殊文字 @ および % は、SNMP コミュニティ ストリングでは使用できません。
- デフォルトの SNMP PDU 値は 1500 バイトです。SNMP エージェントは、1500 バイトを超える応答 PDU をドロップするので、SNMP リクエストは失敗します。1500 バイトを超える MIB データ値を受信するには、**snmp-server packetsize** <*byte-count*>コマンドを使用し

て、パケット サイズを再設定します。有効なバイト数の範囲は 484 ー 17382 です。GETBULK応答がパケットサイズを超えると、データが切り捨てられることがあります。

- スイッチの機能を設定するには、CLIまたはSNMPを使用する必要があります。スイッチに、両方のインターフェイスを使用して機能を設定しないでください。
- シャーシにファンが装着されていない個々のファン OID ツリーでcefcFanTrayOperStatus snmpwalk を使用すると、ツリー内の次の OID エントリに対する応答が返されることがあります。この動作を防ぐには、*snmpwalk* で -CI オプションを使用します。

  この動作は、親OIDをポーリングする場合、またはgetmanyを使用する場合には見られません。
- Cisco Nexus 9000 シリーズ スイッチは、*snmpwalk* 要求に対して最大 10000 個のフラッシュ ファイルをサポートします。
- SNMP トラップが完全で適切な機能動作を実行するには、少なくとも 1 つの実行中の BGP インスタンスが必要です。snmp-server traps 関連のコマンドを設定する前に、BGP ルーティングインスタンスを設定します。
- リリース 10.1(1) 以降、AES-128 は強力な暗号化アルゴリズムであるため、推奨される暗号化アルゴリズムです。ただし、DES 暗号化もサポートされています。

  ダウングレード：DES プライバシー プロトコルを持つユーザが SNMP データベースに存在する場合、**install all** コマンドによる In-Service System Downgrade（ISSD）が中断されます。ユーザは（デフォルトの AES-128 を使用して）再設定または削除する必要があります。コールド リブートの場合、DES を持つ SNMP ユーザは削除されます。
- Cisco NX-OS リリース 10.5（2）以降、ユーザーは、AES-256 を SNMPv3 のプライバシー プロトコルとして構成できます。
  - 以前のリリースにダウングレードする前に、暗号化AES-256 を使用する既存のユーザーを AES-128 に再構成するか、暗号化化 AES-256 を使用するユーザーを削除します。
  - この機能は、すべての N9K プラットフォームでサポートされています。
- SNMPユーザーの構成後にエンジン識別子を構成する場合は、次のアクションを実行してください。
  - エンジン識別子を変更した後、SNMPユーザーと、グループ、ACLを含む関連構成をパスワードとともに再構成します。これにより、認証の失敗と、ユーザーに関連付けられた ACL およびグループへの影響が回避できます。
- Cisco NX-OS リリース 10.3(1)F 以降 、SNMP（MIB – 400G Optic MIB、スイッチ MIB、データパス MIB、インターフェイス MIB）が Cisco Nexus 9808 プラットフォーム スイッチでサポートされています。
- SVI 統計情報は、SNMP キャッシュに対して 120 秒ごとにのみポーリングされます。

- Cisco NX-OS リリース 10.3(3)F 以降では、SNMPv3 ユーザー パスワードのタイプ 6 暗号化が次の制限付きでサポートされています。
  - タイプ 6 暗号化は、次の点に注意した場合にのみ成功します。
    - **feature password encryption aes {tam}** がイネーブルになっていること。
    - プライマリ キーが構成されていること。
    - **pwd_type 6** オプションは、SNMPv3 ユーザーの構成時に指定されます。
  - プライマリ キーの構成を変更すると、SNMP はデータベースに保存されているすべてのタイプ 6 ユーザーを再暗号化します。ただし、SNMP 機能は以前と同じように動作します。
  - プライマリ キーの設定は、スイッチに対してローカルです。ユーザーが 1 つのスイッチからタイプ 6 で構成された実行データを取得し、別のプライマリ キーが構成されている別のスイッチに適用すると、同じユーザーの SNMP 機能が別のスイッチでは動作しない可能性があります。
  - タイプ 6 が設定されている場合は、タイプ 6 がサポートされていないリリースにダウングレードする前に、構成を削除するか、タイプ 6 オプションを再構成してください。
  - ISSU の場合、以前のイメージ（localizedkey、localizedV2key 構成が存在する）からタイプ 6 暗号化がサポートされている新しいイメージに移行すると、SNMP は既存のキーをタイプ 6 暗号化に変換しません。
  - 既存の SALT 暗号化からタイプ 6 暗号化への変換は、**encryption re-encrypt obfuscated** コマンドを使用してサポートされます。
  - 中断を伴うアップグレードや **reload-ascii** コマンドによる ASCII ベースのリロードを実行すると、プライマリ キーが失われ、タイプ 6 ユーザーの SNMP 機能に影響を与えます。
  - ユーザーが **encryption re-encrypt obfuscated** コマンドを使用して再暗号化を強制すると、SNMP はタイプ 6 以外の SNMP ユーザーからのすべてのパスワードをタイプ 6 モードに暗号化します。

（注） SNMP は **encryption delete type6** コマンドをサポートしていません。同じことを示す syslog 警告メッセージも表示されます。

- Cisco NX-OS リリース 10.4(1)F 以降では、SNMP を使用して Electronic Programmable Logic Device（EPLD）ファームウェア バージョンを表示できます。エンティティ MIB 構造の一部として、ファームウェア バージョン、EPLD デバイスのタイプ（IO または MI FPGA）、および EPLD デバイスの親エンティティ（スーパーバイザ、ライン カード、ライン カード拡張モジュール（LEM）など）を表示できます。この機能は、Cisco Nexus 9300-FX/FX2/FX3/GX プラットフォーム スイッチ、N9K-C9332D-H2R スイッチ、および

Nexus 9508 スイッチでサポートされています。Cisco NX-OS リリース 10.4(3)F 以降、この機能は N9K-C9364C-H1 スイッチでもサポートされます。Cisco NX-OS リリース 10.5(2)F 以降、この機能は N9K-X9736C-FX3 ラインカードでもサポートされます。

- Cisco NX-OS リリース 10.4(1)F 以降 、SNMP（MIB – 400G Optic MIB、スイッチ MIB、データパス MIB、インターフェイス MIB）は、次のライン カードとスイッチでサポートされています。
  - Cisco Nexus 9804 スイッチ
  - Cisco Nexus 9332D-H2R スイッチ
  - Cisco Nexus X98900CD-A および X9836DM-A ラインカードと Cisco Nexus 9808 および 9804 スイッチ
- Cisco NX-OS リリース 10.4(2)F 以降、SNMP（MIB – 400G Optic MIB、スイッチ MIB、データパス MIB、インターフェイス MIB）が Cisco Nexus 93400LD-H1 プラットフォーム スイッチでサポートされます。
- Cisco NX-OS リリース 10.4(2)F 以降 、SNMP（MIB – 400G Optic MIB、スイッチ MIB、データパス MIB、インターフェイス MIB）が Cisco Nexus N9K-C9364C-H1 プラットフォーム スイッチでサポートされます。

SNMP のデフォルト設定

次の表に、SNMP パラメータのデフォルト設定を示します。

パラメータ	デフォルト
ライセンス通知	有効（Enabled）

SNMP の設定

（注） この機能の Cisco NX-OS コマンドは、Cisco IOS のコマンドとは異なる場合があるので注意してください。

（注） Cisco NX-OS リリース 9.3(7) 以降、HMAC-SHA-256 認証プロトコルは SNMPv3 に使用されます。

SNMP ユーザの設定

SNMP ユーザを設定できます。

SUMMARY STEPS

1. **configure terminal**
2. **snmp-server user** *name* [**pwd_type 6**] [**auth** {**md5** | **sha** | **sha-224** | **sha-256** | **sha-384** | **sha-512**} *passphrase* [**auto**] [**priv [aes-128] [aes-256]** *passphrase*] [**engineID** *id*] [**localizedkey**] | [**localizedV2key**]]
3. (Optional) **show snmp user**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

ステップ 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	**snmp-server user** *name* [**pwd_type 6**] [**auth** {**md5** | **sha** | **sha-224** | **sha-256** | **sha-384** | **sha-512**} *passphrase* [**auto**] [**priv [aes-128] [aes-256]** *passphrase*] [**engineID** *id*] [**localizedkey**] | [**localizedV2key**]] **Example:** `switch(config)# snmp-server user Admin pwd_type 6 auth sha abcd1234 priv abcdefgh`	認証およびプライバシー パラメータのある SNMP ユーザを設定します。パスフレーズには最大 64 文字の英数字を使用できます。大文字と小文字が区別されます。 **localizedkey** キーワードを使用する場合は、パスフレーズに大文字と小文字を区別した英数字を 130 文字まで使用できます。 **sha** ：認証に HMAC SHA-1 アルゴリズムを使用します。 **sha-224** ：認証に HMAC SHA-224 アルゴリズムを使用します。 **sha-256** ：認証に HMAC SHA-256 アルゴリズムを使用します。 **sha-384** ：認証に HMAC SHA-384 アルゴリズムを使用します。 **sha-512** ：認証に HMAC SHA-512 アルゴリズムを使用します。 **localizedkey** - **localizedkey**キーワードを使用する場合は、パスフレーズに大文字と小文字を区別した英数字を 130 文字まで使用できます。[プレーン テキスト パスワードの代わりに、localizedkey キーワー

	Command or Action	Purpose
		ドを使用して**ハッシュされた** パスワード（**show running config**コマンドからコピーするか、snmpv3 ベースのオープン ソース ハッシュ ジェネレーター ツールを使用してオフラインで生成したもの、ハッシュ化されたパスワードをオフラインで生成する, on page 16を参照）を構成できます。 **Note** ローカライズされたキーを使用する場合は、ハッシュ値の前に `0x` を追加します（例: `0x84a716329158a97ac9f22780629bc26c`）。 **localizedV2key** - localizedV2key キーを使用する場合、パスフレーズは大文字と小文字を区別した、最大 130 文字の英数字文字列にすることができます。先頭に 0x を付ける必要はありません。これは暗号化されたデータであり、オフラインでは生成できないため、**show run** コマンドを使用して localizedv2key を収集します。 engineID の形式は、12 桁のコロンで区切った 10 進数字です。 **Note** • Cisco NX-OS リリース 10.1(1) 以降、AES-128 は SNMPv3 のデフォルトのプライバシー プロトコルです。 • Cisco NX-OS リリース 10.5（2）以降、ユーザーは、AES-256 を SNMPv3 のプライバシー プロトコルとして構成できます。 • Cisco NX-OS リリース 10.3(3)F 以降では、SNMP ユーザー パスワードにタイプ 6 暗号化を提供するために **pwd_type 6** キーワードがサポートされています。
ステップ 3	(Optional) **show snmp user** **Example:** `switch(config)# show snmp user`	1 人または複数の SNMP ユーザに関する情報を表示します。
ステップ 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

ハッシュ化されたパスワードをオフラインで生成する

snmpv3 ベースのオープン ソース ハッシュ ジェネレータ ツールを使用して、ハッシュ化されたパスワードをオフラインで生成する手順は、次のとおりです。

（注） 例としてい挙げられている ID はサンプルの ID で、手順を説明するためだけのものです。

1. スイッチから SNMP engineID を取得します。

   switch# **show snmp engineID**

   サンプル出力：

   ```
   Local SNMP engineID: [Hex] 8000000903D4C93CEA31CC
   [Dec] 128:000:000:009:003:212:201:060:234:049:204
   ```

2. SNMPv3 ベースのオープン ソース ハッシュ ジェネレータを使用して、ハッシュ化されたパスワードをオフラインで生成します。

   Linux$ **snmpv3-hashgen --auth** Hello123 --engine 8000000903D4C93CEA31CC --user1 **--mode priv --hash md5**

   サンプル出力：

   ```
   User: user1
   Auth: Hello123 / 84a716329158a97ac9f22780629bc26c
   Priv: Hello123 / 84a716329158a97ac9f22780629bc26c
   Engine: 8000000903D4C93CEA31CC
   ESXi USM String:
   u1/84a716329158a97ac9f22780629bc26c/84a716329158a97ac9f22780629bc26c/priv
   ```

3. auth および priv の値を使用して、スイッチのパスワードを構成します。

   snmp-server user user1 **auth md5** 0x84a716329158a97ac9f22780629bc26c **priv des 0x**84a716329158a97ac9f22780629bc26c **localizedkey**

SNMP メッセージ暗号化の適用

着信要求に認証または暗号化が必要となるよう SNMP を設定できます。デフォルトでは、SNMP エージェントは認証および暗号化を行わないでも SNMPv3 メッセージを受け付けます。プライバシーを適用する場合、Cisco NX-OS は、**noAuthNoPriv** または **authNoPriv** のいずれかのセキュリティ レベル パラメータを使用するすべての SNMPv3 PDU 要求に対して、許可エラーで応答します。

手順の概要

1. **configure terminal**
2. **snmp-server user** *name* **enforcePriv**
3. **snmp-server globalEnforcePriv**
4. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**snmp-server user** *name* **enforcePriv** 例： `switch(config)# snmp-server user Admin enforcePriv`	このユーザに対してSNMPメッセージ暗号化を適用します。
ステップ 3	**snmp-server globalEnforcePriv** 例： `switch(config)# snmp-server globalEnforcePriv`	すべてのユーザに対してSNMPメッセージ暗号化を適用します。
ステップ 4	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

SNMPv3 ユーザに対する複数のロールの割り当て

SNMP ユーザを作成した後で、そのユーザに複数のロールを割り当てることができます。

（注） 他のユーザにロールを割り当てることができるのは、network-admin ロールに属するユーザだけです。

手順の概要

1. **configure terminal**
2. **snmp-server user** *name group*
3. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	**snmp-server user** *name group* 例： `switch(config)# snmp-server user Admin superuser`	この SNMP ユーザと設定されたユーザ ロールをアソシエートします。
ステップ 3	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

SNMP コミュニティの作成

SNMPv1 または SNMPv2c の SNMP コミュニティを作成できます。

手順の概要

1. **configure terminal**
2. **snmp-server community** *name* {**group** *group* | **ro** | **rw**}
3. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	**snmp-server community** *name* {**group** *group*	**ro**	**rw**} 例： `switch(config)# snmp-server community public ro`	SNMP コミュニティ ストリングを作成します。

	コマンドまたはアクション	目的
ステップ 3	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

SNMP 要求のフィルタリング

アクセス コントロール リスト（ACL）を SNMPv2 コミュニティに割り当てて、SNMP 要求にフィルタを適用できます。割り当てた ACL により着信要求パケットが許可される場合、SNMP はその要求を処理します。ACL により要求が拒否される場合、SNMP はその要求を廃棄して、システム メッセージを送信します。

ACL は次のパラメータで作成します。

- 送信元 IP アドレス
- 宛先 IP アドレス
- 送信元ポート
- 宛先ポート
- プロトコル（UDP または TCP）

手順の概要

1. **configure terminal**
2. **snmp-server community** *name* [**use-ipv4acl** *acl-name*]
3. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	**snmp-server community** *name* [**use-ipv4acl** *acl-name*] 例： `switch(config)# snmp-server community public use-ipv4acl myacl`	SNMP コミュニティに IPv4 ACL ACL を割り当てて SNMPv2 要求をフィルタします。

	コマンドまたはアクション	目的
ステップ 3	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

SNMP 通知レシーバの設定

複数のホスト レシーバーに対して SNMP 通知を生成するよう Cisco NX-OSを設定できます。

手順の概要

1. **configure terminal**
2. **snmp-server host** *ip-address* **traps version 1** *community* [**udp_port** *number*]
3. **snmp-server host** *ip-address* {**traps** | **informs**} **version 2c** *community* [**udp_port** *number*]
4. **snmp-server host** *ip-address* {**traps** | **informs**} **version 3** {**auth** | **noauth** | **priv**} *username* [**udp_port** *number*]
5. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	**snmp-server host** *ip-address* **traps version 1** *community* [**udp_port** *number*] 例： `switch(config)# snmp-server host 192.0.2.1 traps version 1 public`	SNMPv1 トラップのホスト レシーバを設定します。*ip-address* は IPv4 または IPv6 アドレスを使用できます。コミュニティは、最大 255 文字の英数字で指定できます。UDP ポート番号の範囲は 0 ～ 65535 です。
ステップ 3	**snmp-server host** *ip-address* {**traps**	**informs**} **version 2c** *community* [**udp_port** *number*] 例： `switch(config)# snmp-server host 192.0.2.1 informs version 2c public`	SNMPv2c トラップまたはインフォームのホスト レシーバを設定します。*ip-address* は IPv4 または IPv6 アドレスを使用できます。コミュニティは、最大 255 文字の英数字で指定できます。UDP ポート番号の範囲は 0 ～ 65535 です。
ステップ 4	**snmp-server host** *ip-address* {**traps**	**informs**} **version 3** {**auth**	**noauth**	**priv**} *username* [**udp_port** *number*] 例：	SNMPv3 トラップまたは応答要求のホスト レシーバを設定します。*ip-address* は IPv4 または IPv6 アドレスを使用できます。ユーザ名は、最大 255 文字の英

	コマンドまたはアクション	目的
	`switch(config)# snmp-server host 192.0.2.1 informs version 3 auth NMS`	数字で指定できます。UDP ポート番号の範囲は 0 ～ 65535 です。 （注） SNMP マネージャは SNMPv3 メッセージを認証して解読するために、Cisco NX-OS デバイスの SNMP engineID に基づいてユーザ クレデンシャル（authKey/PrivKey）を調べる必要があります。
ステップ 5	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

SNMP 通知用の発信元 インターフェイスの設定

通知の送信元 IP アドレスとしてインターフェイスの IP アドレスを使用するよう、SNMP を設定できます。通知が生成される場合、送信元 IP アドレスは、この設定済みインターフェイスの IP アドレスに基づいています。

次のように発信元インターフェイスを設定できます。

- すべての通知が、すべての SNMP 通知レシーバへ送信される。
- すべての通知が、特定の SNMP 通知レシーバへ送信される。このコンフィギュレーションは、グローバル発信元インターフェイスのコンフィギュレーションよりも優先されます。

（注） 発信トラップパケットの送信元インターフェイス IP アドレスを設定すると、デバイスがトラップの送信に同じインターフェイスを使用することが保証されません。送信元インターフェイス IP アドレスは、SNMP トラップの内部で送信元アドレスを定義し、出力インターフェイス アドレスを送信元として接続が開きます。

手順の概要

1. **configure terminal**
2. **snmp-server host** *ip-address* **source-interface** *if-type if-number* **traps version 2c** *name*
3. **snmp-server host** *ip-address* **source-interface** *if-type if-number* **use-vrf** *vrf-name*
4. **snmp-server host** *ip-address* **source-interface** *if-type if-number* [**udp_port** *number*]
5. **snmp-server source-interface** {**traps** | **informs**} *if-type if-number*
6. **show snmp source-interface**

手順の詳細

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	**snmp-server host** *ip-address* **source-interface** *if-type if-number* **traps version 2c** *name* 例： `snmp-server host 192.0.2.1 source-interface ethernet 2/1 traps version 2c public`	（任意）このホストにトラップ メッセージを送信します。 トラップのバージョンには、通知メッセージに使用する SNMP バージョンを指定します。2c は、SNMPv2c が使用されることを示します。
ステップ 3	**snmp-server host** *ip-address* **source-interface** *if-type if-number* **use-vrf** *vrf-name* 例： `snmp-server host 192.0.2.1 source-interface ethernet 2/1 use-vrf default`	特定の VRF を使用してホスト レシーバと通信するように SNMP を設定します。ip-address は IPv4 または IPv6 アドレスにできます。VRF 名は、最大 32 文字の英数字で指定できます。 （注） このコマンドによってホスト設定は削除されません。
ステップ 4	**snmp-server host** *ip-address* **source-interface** *if-type if-number* [**udp_port** *number*] 例： `switch(config)# snmp-server host 192.0.2.1 source-interface ethernet 2/1`	SNMPv2c トラップまたはインフォームのホスト レシーバを設定します。*ip-address* は IPv4 または IPv6 アドレスを使用できます。サポートされているインターフェイス タイプを特定するために「**?**」を使用します。UDP ポート番号の範囲は 0 ～ 65535 です。 このコンフィギュレーションは、グローバル発信元インターフェイスのコンフィギュレーションよりも優先されます。
ステップ 5	**snmp-server source-interface** {**traps** | **informs**} *if-type if-number* 例： `switch(config)# snmp-server source-interface traps ethernet 2/1`	SNMPv2c トラップまたは応答要求を送信するよう発信元インターフェイスを設定します。サポートされているインターフェイス タイプを特定するために「**?**」を使用します。
ステップ 6	**show snmp source-interface** 例： `switch(config)# show snmp source-interface`	設定した発信元インターフェイスの情報を表示します。

通知ターゲット ユーザの設定

SNMPv3 インフォーム通知を通知ホスト レシーバに送信するには、デバイスに通知ターゲット ユーザを設定する必要があります。

Cisco NX-OS は通知ターゲット ユーザのクレデンシャルを使用して、設定された通知ホスト レシーバへの SNMPv3 応答要求通知メッセージを暗号化します。

（注） 受信した INFORM PDU を認証して解読する場合、Cisco NX-OS で設定されているのと同じ、応答要求を認証して解読するユーザ クレデンシャルが通知ホスト レシーバに必要です。

手順の概要

1. **configure terminal**
2. **snmp-server user** *name* [**auth** {**md5** | **sha** | **sha-256**} *passphrase* [**auto**] [**priv** *passphrase*] [**engineID** *id*]
3. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	**snmp-server user** *name* [**auth** {**md5** | **sha** | **sha-256**} *passphrase* [**auto**] [**priv** *passphrase*] [**engineID** *id*] 例： `switch(config)# snmp-server user NMS auth sha abcd1234 priv abcdefgh engineID 00:00:00:63:00:01:00:10:20:15:10:03`	通知ホスト レシーバのエンジン ID を指定して、通知ターゲット ユーザを設定します。エンジン ID の形式は、12 桁のコロンで区切った 10 進数字です。 （注） リリース 10.1(1) 以降、AES-128 は SNMPv3 のデフォルトのプライバシー プロトコルです。
ステップ 3	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

VRF を使用する SNMP 通知レシーバの設定

SNMP 通知レシーバの VRF 到達可能性およびフィルタリング オプションを設定すると、SNMP によって CISCO-SNMP-TARGET-EXT-MIB の cExtSnmpTargetVrfTable にエントリが追加されます。

（注） VRF 到達可能性またはフィルタリング オプションを設定する前に、ホストを設定する必要があります。

ホスト レシーバに到達するように設定した VRF を使用したり、または通知が発生した VRF に基づいて通知をフィルタするように Cisco NX-OS を設定できます。

手順の概要

1. **configure terminal**
2. [**no**] **snmp-server host** *ip-address* **use-vrf** *vrf-name* [**udp_port** *number*]
3. [**no**] **snmp-server host** *ip-address* **filter-vrf** *vrf-name* [**udp_port** *number*]
4. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	[**no**] **snmp-server host** *ip-address* **use-vrf** *vrf-name* [**udp_port** *number*] 例： `switch(config)# snmp-server host 192.0.2.1 use-vrf Blue`	特定の VRF を使用してホスト レシーバと通信するように SNMP を設定します。*ip-address* を IPv4 または IPv6 アドレスにできます。VRF 名には最大 255 の英数字を使用できます。UDP ポート番号の範囲は 0 ～ 65535 です。このコマンドによって、CISCO-SNMP-TARGET-EXT-MB の ExtSnmpTargetVrfTable にエントリが追加されます。 このコマンドの **no** 形式は、設定されたホストの VRF 到達可能性情報を削除し、CISCO-SNMP-TARGET-EXT-MB の ExtSnmpTargetVrfTable からエントリを削除します。 （注） このコマンドによってホスト設定は削除されません。

	コマンドまたはアクション	目的
ステップ 3	[**no**] **snmp-server host** *ip-address* **filter-vrf** *vrf-name* [**udp_port** *number*] 例： `switch(config)# snmp-server host 192.0.2.1 filter-vrf Red`	設定された VRF に基づいて、通知ホスト レシーバへの通知をフィルタリングします。*ip-address* は IPv4 または IPv6 アドレスを使用できます。VRF 名には最大 255 の英数字を使用できます。UDP ポート番号の範囲は 0 ～ 65535 です。 このコマンドによって、CISCO-SNMP-TARGET-EXT-MB の ExtSnmpTargetVrfTable にエントリが追加されます。 このコマンドの **no** 形式は、設定されたホストの VRF フィルタ情報を削除し、CISCO-SNMP-TARGET-EXT-MB の ExtSnmpTargetVrfTable からエントリを削除します。 （注） このコマンドによってホスト設定は削除されません。
ステップ 4	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

帯域内ポートを使用してトラップを送信するための SNMP 設定

帯域内ポートを使用してトラップを送信するよう SNMP を設定できます。このようにするには、（グローバルまたはホスト レベルで）発信元インターフェイスを設定し、トラップを送信するための VRF を設定します。

手順の概要

1. **configure terminal**
2. **snmp-server source-interface traps** *if-type if-number*
3. （任意） **show snmp source-interface**
4. **snmp-server host** *ip-address* **use-vrf** *vrf-name* [**udp_port** *number*]
5. （任意） **show snmp host**
6. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	**snmp-server source-interface traps** *if-type if-number* 例： `switch(config)# snmp-server source-interface traps ethernet 1/2`	SNMP トラップを送信するための発信元インターフェイスをグローバルに設定します。サポートされているインターフェイス タイプを特定するために「?」を使用します。 グローバル レベルまたはホスト レベルで発信元インターフェイスを設定できます。発信元インターフェイスをグローバルに設定すると、新しいホスト コンフィギュレーションはグローバルなコンフィギュレーションを使用してトラップを送信します。 （注） 発信元インターフェイスをホスト レベルで設定するには、**snmp-server host** *ip-address* **source-interface** *if-type if-number* コマンドを使用します。
ステップ 3	（任意） **show snmp source-interface** 例： `switch(config)# show snmp source-interface`	設定した発信元インターフェイスの情報を表示します。
ステップ 4	**snmp-server host** *ip-address* **use-vrf** *vrf-name* [**udp_port** *number*] 例： `switch(config)# snmp-server host 171.71.48.164 use-vrf default`	特定の VRF を使用してホスト レシーバと通信するように SNMP を設定します。*ip-address* は IPv4 または IPv6 アドレスを使用できます。VRF 名には最大 255 の英数字を使用できます。UDP ポート番号の範囲は 0 ～ 65535 です。このコマンドによって、CISCO-SNMP-TARGET-EXT-MB の ExtSnmpTargetVrfTable にエントリが追加されます。 （注） デフォルトでは、SNMP は管理 VRF を使用してトラップを送信します。管理 VRF を使用しない場合は、このコマンドを使用して対象の VRF を指定する必要があります。
ステップ 5	（任意） **show snmp host** 例：	設定した SNMP ホストの情報を表示します。

	コマンドまたはアクション	目的
	`switch(config)# show snmp host`	
ステップ 6	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップコンフィギュレーションにコピーします。

SNMP 通知のイネーブル化

通知をイネーブルまたはディセーブルにできます。通知名を指定しなかった場合、Cisco NX-OSは、BGP、EIGRP、およびOSPFの通知を除き、通知をすべてイネーブルにします。

Note **snmp-server enable traps** コマンドを使用すると、設定通知ホストレシーバによっては、トラップとインフォームの両方をイネーブルにできます。

次の表に、Cisco NX-OS MIB の通知を有効にするコマンドを示します。

***Table 2: SNMP* 通知のイネーブル化**

MIB	関連コマンド
すべての通知（BGP、EIGRP、およびOSPFを除く）	**snmp-server enable traps**
CISCO-AAA-SERVER-MIB	**snmp-server enable traps aaa** **snmp-server enable traps aaa server-state-change**
CISCO-BGP4-MIB	**snmp-server enable traps bgp**
CISCO-CALLHOME-MIB	**snmp-server enable traps callhome** **snmp-server enable traps callhome event-notify** **snmp-server enable traps callhome smtp-send-fail**
CISCO-CONFIG-MAN-MIB	**snmp-server enable traps config** **snmp-server enable traps config ccmCLIRunningConfigChanged**
CISCO-EIGRP-MIB	**snmp-server enable traps eigrp** [*tag*]
CISCO-ERR-DISABLE-MIB	**snmp-server enable traps link cerrDisableInterfaceEventRev1**

MIB	関連コマンド
ENTITY-MIB、CISCO-ENTITY-SENSOR-MIB	**snmp-server enable traps entity** **snmp-server enable traps entity entity_fan_status_change** **snmp-server enable traps entity entity_mib_change** **snmp-server enable traps entity entity_module_inserted** **snmp-server enable traps entity entity_module_removed** **snmp-server enable traps entity entity_module_status_change** **snmp-server enable traps entity entity_power_out_change** **snmp-server enable traps entity entity_power_status_change** **snmp-server enable traps entity entity_unrecognised_module**
CISCO-FEATURE-CONTROL-MIB	**snmp-server enable traps feature-control** **snmp-server enable traps feature-control FeatureOpStatusChange**
CISCO-HSRP-MIB	**snmp-server enable traps hsrp** **snmp-server enable traps hsrp state-change**
IF-MIB	**snmp-server enable traps link** **snmp-server enable traps link extended-linkDown** **snmp-server enable traps link extended-linkUp** **snmp-server enable traps link cieLinkDown** **snmp-server enable traps link cieLinkUp** **snmp-server enable traps link linkDown** **snmp-server enable traps link linkUp**
OSPF-MIB, OSPF-TRAP-MIB	**snmp-server enable traps ospf** [*tag*] **snmp-server enable traps ospf lsa** **snmp-server enable traps ospf rate-limit** *rate*

MIB	関連コマンド
CISCO-RF-MIB	**snmp-server enable traps rf** **snmp-server enable traps rf redundancy_framework**
CISCO-RMON-MIB	**snmp-server enable traps rmon** **snmp-server enable traps rmon fallingAlarm** **snmp-server enable traps rmon hcFallingAlarm** **snmp-server enable traps rmon hcRisingAlarm** **snmp-server enable traps rmon risingAlarm**
SNMPv2-MIB	**snmp-server enable traps snmp** **snmp-server enable traps snmp authentication**
CISCO-MAC-NOTIFICATION-MIB	**snmp-server enable trap link cmn-mac-move-notification**
CISCO-PORT-STORM-CONTROL-MIB	**storm-control action trap**
CISCO-STP-EXTENSIONS-MIB	**snmp-server enable traps stpx stpxMstInconsistencyUpdate**
CISCO-STP-BRIDGE-MIB	**snmp-server enable traps bridge** **snmp-server enable traps bridge newroot** **snmp-server enable traps bridge topologychange**
CISCO-STPX-MIB	**snmp-server enable traps stpx** **snmp-server enable traps stpx inconsistency** **snmp-server enable traps stpx loop-inconsistency** **snmp-server enable traps stpx root-inconsistency**
CISCO-SYSTEM-EXT-MIB	**snmp-server enable traps sysmgr** **snmp-server enable traps sysmgr cseFailSwCoreNotifyExtended**
UPGRADE-MIB	**snmp-server enable traps upgrade** **snmp-server enable traps upgrade UpgradeJobStatusNotify** **snmp-server enable traps upgrade UpgradeOpNotifyOnCompletion**

MIB	関連コマンド
VTP-MIB	**snmp-server enable traps vtp** **snmp-server enable traps vtp notifs** **snmp-server enable traps vtp vlancreate** **snmp-server enable traps vtp vlandelete**

指定した通知を有効にするには、示しているようにコンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
snmp-server enable traps **例:** `switch(config)# snmp-server enable traps`	すべての SNMP 通知をイネーブルにします。
snmp-server enable traps aaa [**server-state-change**] **例:** `switch(config)# snmp-server enable traps aaa`	AAA SNMP 通知をイネーブルにします。任意で、次の特定の通知をイネーブルにします。 • **server-state-change**：AAA サーバの状態変化通知を有効にします。
snmp-server enable traps bgp **例:** `switch(config)# snmp-server enable traps bgp`	ボーダー ゲートウェイ プロトコル（BGP）SNMP 通知を有効にします。
snmp-server enable traps bridge [**newroot**] [**topologychange**] **例:** `switch(config)# snmp-server enable traps bridge`	STP ブリッジ SNMP 通知をイネーブルにします。任意で、次の特定の通知をイネーブルにします。 • **newroot**：STP の新しいルート ブリッジ通知を有効にします。 • **topologychange**：STP ブリッジのトポロジ変更通知を有効にします。
snmp-server enable traps callhome [**event-notify**] [**smtp-send-fail**] **例:** `switch(config)# snmp-server enable traps callhome`	Call Home 通知をイネーブルにします。任意で、次の特定の通知をイネーブルにします。 • **event-notify**：Call Home の外部イベント通知を有効にします。 • **smtp-send-fail**：簡易メール転送プロトコル（SMTP）メッセージの送信失敗通知を有効にします。

コマンド	目的
snmp-server enable traps config [**ccmCLIRunningConfigChanged**] **例:** `switch(config)# snmp-server enable traps config`	コンフィギュレーションの変更に対してSNMP通知をイネーブルにします。 • **ccmCLIRunningConfigChanged**：実行中または起動時のコンフィギュレーションで、コンフィギュレーションの変更に対して SNMP 通知を有効にします。
snmp-server enable traps eigrp [*tag*] **例:** `switch(config)# snmp-server enable traps eigrp`	CISCO-EIGRP-MIB SNMP 通知をイネーブルにします。
snmp-server enable traps entity [**entity_fan_status_change**] [**entity_mib_change**] [**entity_module_inserted**] [**entity_module_removed**] [**entity_module_status_change**] [**entity_power_out_change**] [**entity_power_status_change**] [**entity_unrecognised_module**] **例:** `switch(config)# snmp-server enable traps entity`	ENTITY-MIB SNMP 通知をイネーブルにします。任意で、次の特定の通知をイネーブルにします。 • **entity_fan_status_change**：エンティティファンの状態変化通知を有効にします。 • **entity_mib_change**：エンティティ MIB 変更通知を有効にします。 • **entity_module_inserted**：エンティティ モジュール挿入通知を有効にします。 • **entity_module_removed**：エンティティ モジュール削除通知を有効にします。 • **entity_module_status_change**：エンティティ モジュール ステータス変更通知を有効にします。 • **entity_power_out_change**：エンティティの出力パワー変更通知を有効にします。 • **entity_power_status_change**：エンティティのパワー ステータス変更通知を有効にします。 • **entity_unrecognised_module**：エンティティの未確認モジュール通知を有効にします。

コマンド	目的
snmp-server enable traps feature-control [**FeatureOpStatusChange**] **例:** `switch(config)# snmp-server enable traps feature-control`	機能制御 SNMP 通知をイネーブルにします。任意で、次の特定の通知をイネーブルにします。 • **FeatureOpStatusChange**：機能操作の状態変化通知を有効にします。
snmp-server enable traps hsrp state-change **例:** `switch(config)# snmp-server enable traps hsrp`	CISCO-HSRP-MIB SNMP 通知をイネーブルにします。任意で、次の特定の通知をイネーブルにします。 • **state-change**：HSRP の状態変化通知を有効にします。
snmp-server enable traps license [**notify-license-expiry**] [**notify-license-expiry-warning**] [**notify-licensefile-missing**] [**notify-no-license-for-feature**] **例:** `switch(config)# snmp-server enable traps license`	ENTITY-MIB SNMP 通知をイネーブルにします。任意で、次の特定の通知をイネーブルにします。 • **notify-license-expiry**：ライセンス失効通知を有効にします。 • **notify-license-expiry-warning**：ライセンス失効の警告通知を有効にします。 • **notify-licensefile-missing**：ライセンス ファイル不明通知を有効にします。 • **notify-no-license-for-feature**：no-license-installed-for-feature 通知を有効にします。

コマンド	目的
snmp-server enable traps link [**cieLinkDown**] [**cieLinkUp**] [**cmn-mac-move-notification**] [**IETF-extended-linkDown**] [**IETF-extended-linkUp**] [**cisco-extended-linkDown**] [**cisco-extended-linkUp**][**linkDown**] [**linkUp**] **例:** `switch(config)# snmp-server enable traps link`	IF-MIB リンク通知をイネーブルにします。任意で、以下の特定の通知をイネーブルにします。 • **IETF-extended-linkDown**：Cisco 拡張リンク ステート ダウン通知をイネーブルにします。 • **IETF-extended-linkUp**：Cisco 拡張リンク ステート アップ通知をイネーブルにします。 • **cmn-mac-move-notification**：MACアドレス移動通知をイネーブルにします。 • **cisco-extended-linkDown**—：Internet Engineering Task Force（インターネット エンジニアリング タスク フォース、IETF）の拡張リンク ステート ダウン通知をイネーブルにします。 • **cisco-extended-linkUP**：Internet Engineering Task Force（IETF）の拡張リンク ステート アップ通知をイネーブルにします。 • **linkDown**：IETF リンク ステート ダウン通知を有効にします。 • **linkUp**：IETF リンク ステート アップ通知を有効にします。
snmp-server enable traps ospf [*tag*] [**lsa**] **例:** `switch(config)# snmp-server enable traps ospf`	Open Shortest Path First（OSPF）通知を有効にします。任意で、次の特定の通知をイネーブルにします。 • **lsa**：OSPF リンク ステート アドバタイズメント（LSA）通知を有効にします。
snmp-server enable traps rf [**redundancy-framework**] **例:** `switch(config)# snmp-server enable traps rf`	冗長フレームワーク（RF）SNMP通知をイネーブルにします。任意で、次の特定の通知をイネーブルにします。 • **redundancy-framework**：RF スーパーバイザ スイッチオーバー MIB 通知を有効にします。

コマンド	目的
snmp-server enable traps rmon [**fallingAlarm**] [**hcFallingAlarm**] [**hcRisingAlarm**] [**risingAlarm**] **例:** `switch(config)# snmp-server enable traps rmon`	リモート モニタリング（RMON）SNMP 通知をイネーブルにします。任意で、次の特定の通知をイネーブルにします。 • **fallingAlarm**：RMON 下限アラーム通知を有効にします。 • **hcFallingAlarm**：RMON high-capacity 下限アラーム通知を有効にします。 • **hcRisingAlarm**：RMON high-capacity 上限アラーム通知を有効にします。 • **risingAlarm**：RMON 上限アラーム通知を有効にします。
snmp-server enable traps snmp [**authentication**] **例:** `switch(config)# snmp-server enable traps snmp`	一般的な SNMP 通知をイネーブルにします。任意で、次の特定の通知をイネーブルにします。 • **authentication**：SNMP 認証通知を有効にします。
snmp-server enable traps stpx[**inconsistency**] [**loop-inconsistency**] [**root-inconsistency**] **例:** `switch(config)# snmp-server enable traps stpx`	SNMP STPX 通知を有効にします。任意で、次の特定の通知をイネーブルにします。 • **inconsistency**：SNMP STPX MIB 不一致アップデート通知を有効にします。 • **loop-inconsistency**：SNMP STPX MIB ループ不一致アップデート通知を有効にします。 • **root-inconsistency**：SNMP STPX MIB ルート不一致アップデート通知を有効にします。
snmp-server enable traps syslog [**message-generated**] **例:** `switch(config)# snmp-server enable traps syslog`	定義された SNMP ホストに syslog メッセージをトラップとして送信します。任意で、次の特定の通知をイネーブルにします。 • **message-generate**：ソフトウェア ログ メッセージ生成通知を有効にします。

コマンド	目的
snmp-server enable traps sysmgr [**cseFailSwCoreNotifyExtended**] **例:** `switch(config)# snmp-server enable traps sysmgr`	ソフトウェア変更通知をイネーブルにします。任意で、次の特定の通知をイネーブルにします。 • **cseFailSwCoreNotifyExtended**：ソフトウェア コア通知を有効にします。
snmp-server enable traps upgrade [**UpgradeJobStatusNotify**] [**UpgradeOpNotifyOnCompletion**] **例:** `switch(config)# snmp-server enable traps upgrade`	アップグレード通知をイネーブルにします。任意で、次の特定の通知をイネーブルにします。 • **UpgradeJobStatusNotify**：アップグレード ジョブ ステータス通知を有効にします。 • **UpgradeOpNotifyOnCompletion**：アップグレード グローバル ステータス通知を有効にします。
snmp-server enable traps vtp[**notifs**] [**vlancreate**] [**vlandelete**] **例:** `switch(config)# snmp-server enable traps vtp`	VTP 通知を有効にします。任意で、次の特定の通知をイネーブルにします。 • **notifs**：VTP 通知を有効にします。 • **vlancreate**：VLAN 作成の通知を有効にします。 • **vlandelete**：VLAN 削除の通知を有効にします。
storm-control action traps **例:** `switch(config-if)# storm-control action traps`	トラフィック ストーム制御の制限に達した場合のトラフィック ストーム制御通知を有効にします。

インターフェイスでのリンク通知のディセーブル化

個別のインターフェイスで linkUp および linkDown 通知をディセーブルにできます。フラッピング インターフェイス（Up と Down の間を頻繁に切り替わるインターフェイス）で、この制限通知を使用できます。

手順の概要

1. **configure terminal**
2. **interface** *type slot*/*port*
3. **no snmp trap link-status**

4. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**interface** *type slot*/*port* 例： `switch(config)# interface ethernet 2/2`	インターフェイスの SNMP リンクステート トラップをディセーブルにします。このコマンドは、デフォルトでイネーブルになっています。
ステップ 3	**no snmp trap link-status** 例： `switch(config-if)# no snmp trap link-status`	インターフェイスの SNMP リンクステート トラップをディセーブルにします。このコマンドは、デフォルトでイネーブルになっています。
ステップ 4	（任意） **copy running-config startup-config** 例： `switch(config-if)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

インターフェイスの SNMP ifIndex の表示

SNMP ifIndex は、関連するインターフェイス情報をリンクするために複数の SNMP MIB にわたって使用されます。

手順の概要

1. **show interface snmp-ifindex**

手順の詳細

手順

	コマンドまたはアクション	目的

ステップ 1	**show interface snmp-ifindex**
	例：
	`switch# show interface snmp-ifindex | grep -i Eth12/1`
	`Eth12/1 441974784 (0x1a580000)`
		すべてのインターフェイスについて、IF-MIB から永続的な SNMP ifIndex 値を表示します。任意で、**|** キーワードと **grep** キーワードを使用すると、出力で特定のインターフェイスを検索できます。

	コマンドまたはアクション	目的

TCP による SNMP のワンタイム認証の有効化

TCP セッション上で SNMP に対するワンタイム認証をイネーブルにできます。

手順の概要

1. **configure terminal**
2. **snmp-server tcp-session** [**auth**]
3. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

ステップ 2	**snmp-server tcp-session** [**auth**] 例： `switch(config)# snmp-server tcp-session`	TCP セッション上で SNMP に対するワンタイム認証をイネーブルにします。デフォルトではディセーブルになっています。
ステップ 3	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップコンフィギュレーションにコピーします。

SNMP スイッチのコンタクト（連絡先）およびロケーション情報の指定

32 文字までの長さで（スペースを含まない）デバイスのコンタクト情報とデバイスのロケーションを指定できます。

SUMMARY STEPS

1. **configure terminal**
2. **snmp-server contact** *name*
3. **snmp-server location** *name*
4. (Optional) **show snmp**

5. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

ステップ 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**snmp-server contact** *name* **Example:** `switch(config)# snmp-server contact Admin`	SNMP コンタクト名として sysContact を設定します。 **Note** *name* パラメータは必須です。ただし、**no** 形式を使用した コマンドの設定解除の間、このパラメータはオプションです。
ステップ 3	**snmp-server location** *name* **Example:** `switch(config)# snmp-server location Lab-7`	SNMP ロケーションとして sysLocation を設定します。 **Note** *name* パラメータは必須です。ただし、**no** 形式を使用した コマンドの設定解除の間、このパラメータはオプションです。
ステップ 4	(Optional) **show snmp** **Example:** `switch(config)# show snmp`	1 つまたは複数の宛先プロファイルに関する情報を表示します。
ステップ 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

コンテキストとネットワーク エンティティ間のマッピング設定

プロトコル インスタンス、VRF などの論理ネットワーク エンティティに対する SNMP コンテキストのマッピングを設定できます。

Before you begin

論理ネットワーク エンティティのインスタンスを決定します。VRF およびプロトコル インスタンスの詳細については、『Cisco Nexus 9000 シリーズ NX-OS ユニキャスト ルーティング設

定ガイド』または『Cisco Nexus 9000 シリーズ NX-OS マルチキャスト ルーティング設定ガイド』を参照してください。

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **snmp-server context** *context-name* [**instance** *instance-name*] [**vrf** *vrf-name*] [**topology** *topology-name*]
3. (Optional) **snmp-server mib community-map** *community-name* **context** *context-name*
4. (Optional) **show snmp context**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

ステップ 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	[**no**] **snmp-server context** *context-name* [**instance** *instance-name*] [**vrf** *vrf-name*] [**topology** *topology-name*] **Example:** `switch(config)# snmp-server context public1 vrf red`	SNMP コンテキストをプロトコル インスタンス、VRF、またはトポロジにマッピングします。名前には最大 32 の英数字を使用できます。 **no** オプションは、SNMP コンテキストとプロトコル インスタンス、VRF、またはトポロジ間のマッピングを削除します。 **Note** コンテキスト マッピングを削除する目的で、インスタンス、VRF、またはトポロジを入力しないでください。インスタンス、vrf、またはトポロジ キーワードを使用すると、コンテキストとゼロ長ストリング間のマッピングが設定されます。
ステップ 3	(Optional) **snmp-server mib community-map** *community-name* **context** *context-name* **Example:** `switch(config)# snmp-server mib community-map public context public1`	SNMPv2c コミュニティを SNMP コンテキストにマッピングします。名前には最大 32 の英数字を使用できます。
ステップ 4	(Optional) **show snmp context** **Example:** `switch(config)# show snmp context`	1 つまたは複数の SNMP コンテキストに関する情報を表示します。

	Command or Action	Purpose
ステップ 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

SNMP のディセーブル化

デバイスの SNMP を無効にできます。

手順の概要

1. **configure terminal**
2. **no snmp-server protocol enable**

手順の詳細

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	**no snmp-server protocol enable** 例： `switch(config)# no snmp-server protocol enable`	SNMP をディセーブルにします。SNMP はデフォルトでイネーブルになっています。 （注） SNMPv2 を無効にせずに SNMPv1 を無効にすることはできません。SNMPv1 を無効にする場合は、SNMPv3 のみを設定するか、SNMP を完全に無効にします。

SNMP サーバ カウンタ キャッシュ更新タイマーの管理

Cisco NX-OS がキャッシュ ポートの状態を保持する時間は、秒単位で変更できます。

手順の概要

1. **configure terminal**
2. **snmp-server counter cache timeout** *seconds*
3. （任意） **show running-config snmp all |i cac**
4. **no snmp-server counter cache enable**

手順の詳細

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**snmp-server counter cache timeout** *seconds* 例： `switch(config)# snmp-server counter cache timeout 1200`	ポートの状態がローカル キャッシュに保持される時間を秒単位で定義します。カウンタ キャッシュはデフォルトで有効になっており、デフォルトのキャッシュ タイムアウト値は 10 秒です。無効にすると、デフォルトのキャッシュ タイムアウト値は 50 秒になります。範囲は 1 ～ 3600 です。 （注） End of Row（EoR）スイッチングの場合、範囲は 10 ～ 3600 です。
ステップ 3	（任意） **show running-config snmp all |i cac** 例： `switch(config)# copy running-config snmp all | i cac`	設定された SNMP サーバ カウンタ キャッシュ更新タイムアウト値を表示します。
ステップ 4	**no snmp-server counter cache enable** 例： `switch(config)# no snmp-server counter cache enable`	カウンタ キャッシュの更新を無効にします。 （注） カウンタ キャッシュの更新が無効になっている場合、timeout パラメータに設定された値によって、ポートの状態がカウンタ キャッシュに保持される時間が決まります。

AAA 同期時間の変更

同期したユーザ設定を Cisco NX-OS に維持させる時間の長さを変更できます。

手順の概要

1. **configure terminal**
2. **snmp-server aaa-user cache-timeout** *seconds*
3. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**snmp-server aaa-user cache-timeout** *seconds* 例： `switch(config)# snmp-server aaa-user cache-timeout 1200`	ローカル キャッシュで AAA 同期ユーザ設定を維持する時間を設定します。値の範囲は 1 ～ 86400 秒です。デフォルトは 3600 です。
ステップ 3	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

SNMP ローカル エンジン ID の設定

Cisco NX-OS リリース 7.0(3)I6(1) 以降では、ローカルデバイスにエンジン ID を設定できます。

Note SNMP ローカル エンジン ID を設定すると、すべての SNMP ユーザ、V3 ユーザに設定されたホスト、およびコミュニティ ストリングを再設定する必要があります。Cisco NX-OS リリース 7.0(3)I7(1) 以降では、SNMP ユーザとコミュニティ ストリングのみを再設定する必要があります。

SUMMARY STEPS

1. **configure terminal**
2. **snmp-server engineID local** *engineid-string*
3. **show snmp engineID**
4. [**no**] **snmp-server engineID local** *engineid-string*
5. **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

ステップ 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**snmp-server engineID local** *engineid-string* **Example:** `switch(config)# snmp-server engineID local AA:BB:CC:1A:2C:10`	ローカルデバイスの SNMP エンジン ID を変更します。 ローカル エンジン ID は、コロンで指定された 16 進数オクテットのリストとして設定する必要があります。ここでは 10 ～ 64 の範囲の偶数 16 進数文字が使用され、2 つの 16 進数文字ごとにコロンで区切られます。たとえば、80:00:02:b8:04:61:62:63 です。
ステップ 3	**show snmp engineID** **Example:** `switch(config)# show snmp engineID`	設定されている SNMP エンジンの ID を表示します。
ステップ 4	[**no**] **snmp-server engineID local** *engineid-string* **Example:** `switch(config)# no snmp-server engineID local AA:BB:CC:1A:2C:10`	ローカル エンジン ID を無効にし、自動生成されたデフォルトのエンジン ID を設定します。
ステップ 5	Required: **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

SNMP の設定の確認

SNMP 設定情報を表示するには、次の作業を行います。

コマンド	目的
show interface snmp-ifindex	すべてのインターフェイスについて（IF-MIB から）SNMP の ifIndex 値を表示します。

コマンド	目的
show running-config snmp [**all**]	SNMP の実行コンフィギュレーションを表示します。 10.1(1) より前のリリースから 10.1(1) に導入された SNMP ユーザは、設定されたプライバシー プロトコル AES-128 または DES で表示されます。新しいユーザ（リリース 10.1(1) 以降）は、デフォルトで AES-128 プロトコルで設定されます。 9.3(8) リリース以降、show run の SNMPv3 ユーザは、ハッシュではなく SALT 形式で表示されます。
show snmp	SNMP ステータスを表示します。
show snmp community	SNMP コミュニティ ストリングを表示します。 **Note** **snmp-server mib community-map** コマンドの SNMP コンテキストの名前が 11 文字を超える場合、**show snmp community** コマンドの出力は表形式ではなく垂直形式で表示されます。
show snmp context	SNMP コンテキスト マッピングを表示します。
show snmp engineID	SNMP engineID を表示します。
show snmp group	SNMP ロールを表示します。
show snmp host	設定した SNMP ホストの情報を表示します。
show snmp session	SNMP セッションを表示します。

コマンド	目的
show snmp source-interface	設定した発信元インターフェイスの情報を表示します。
show snmp trap	イネーブルまたはディセーブルである SNMP 通知を表示します。
show snmp user	SNMPv3 ユーザを表示します。

SNMP エンティティ

次の SNMP エンティティ MIB OID を使用して、SNMP EPLD 情報を表示できます。

1. entPhysicalName：1.3.6.1.2.1.47.1.1.1.1.7
2. entPhysicalDescr：1.3.6.1.2.1.47.1.1.1.1.2
3. entPhysicalContainedIn : 1.3.6.1.2.1.47.1.1.1.1.4
4. entPhysicalFirmwareRev : 1.3.6.1.2.1.47.1.1.1.1.9

次に、出力例を示します。

```
bgl-ads-4144:167> snmpget -v 2c -c Cisco_59485 10.197.137.54
.1.3.6.1.2.1.47.1.1.1.1.2.120408 << entPhysicalDescr
SNMPv2-SMI::mib-2.47.1.1.1.1.2.120408 = STRING: "Module-1 IO FPGA"
bgl-ads-4144:168> snmpget -v 2c -c Cisco_59485 10.197.137.54
.1.3.6.1.2.1.47.1.1.1.1.4.120408 << entPhysicalContainedIn
SNMPv2-SMI::mib-2.47.1.1.1.1.4.120408 = INTEGER: 22
bgl-ads-4144:169> snmpget -v 2c -c Cisco_59485 10.197.137.54
.1.3.6.1.2.1.47.1.1.1.1.7.120408 << entPhysicalName
SNMPv2-SMI::mib-2.47.1.1.1.1.7.120408 = STRING: "Module-1 IO FPGA"
bgl-ads-4144:170> snmpget -v 2c -c Cisco_59485 10.197.137.54
.1.3.6.1.2.1.47.1.1.1.1.9.120408 << entPhysicalFirmwareRev
SNMPv2-SMI::mib-2.47.1.1.1.1.9.120408 = STRING: "0x12"
```

SNMP の設定例

次に、Blue VRF を使用して、ある通知ホスト レシーバに Cisco linkUp または Down 通知を送信するよう Cisco NX-OS を設定し、Admin と NMS という 2 つの SNMP ユーザを定義する例を示します。

```
configure terminal
snmp-server contact Admin@company.com
snmp-server user Admin auth sha abcd1234 priv abcdefgh
snmp-server user NMS auth sha abcd1234 priv abcdefgh engineID
00:00:00:63:00:01:00:22:32:15:10:03
snmp-server host 192.0.2.1 informs version 3 auth NMS
snmp-server host 192.0.2.1 use-vrf Blue
```

```
snmp-server enable traps link cisco
```

次に、ホスト レベルで設定された帯域内ポートを使用してトラップを送信するよう、SNMP を設定する例を示します。

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# snmp-server host 171.71.48.164 version 2c public
switch(config)# snmp-server host 171.71.48.164 source-interface ethernet 1/2
switch(config)# show snmp host
-------------------------------------------------------------------
Host Port Version Level Type SecName
-------------------------------------------------------------------
171.71.48.164 162 v2c noauth trap public
Source interface: Ethernet 1/2
-------------------------------------------------------------------
switch(config)# snmp-server host 171.71.48.164 use-vrf default
switch(config)# show snmp host
-------------------------------------------------------------------
Host Port Version Level Type SecName
-------------------------------------------------------------------
171.71.48.164 162 v2c noauth trap public
Use VRF: default
Source interface: Ethernet 1/2
-------------------------------------------------------------------
```

次に、グローバルに設定した帯域内ポートを使用してトラップを送信するよう SNMP を設定する例を示します。

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# snmp-server source-interface traps ethernet 1/2
switch(config)# show snmp source-interface
-------------------------------------------------------------------
Notification source-interface
-------------------------------------------------------------------
trap Ethernet1/2
inform -
-------------------------------------------------------------------
switch(config)# snmp-server host 171.71.48.164 use_vrf default
switch(config)# show snmp host
-------------------------------------------------------------------
Host Port Version Level Type SecName
-------------------------------------------------------------------
171.71.48.164 162 v2c noauth trap public
Use VRF: default
Source interface: Ethernet 1/2
-------------------------------------------------------------------
```

VRF red を SNMPv2c のパブリック コミュニティ ストリングにマッピングする例を示します。

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# vrf context red
switch(config-vrf)# exit
switch(config)# snmp-server context public1 vrf red
switch(config)# snmp-server mib community-map public context public1
```

OSPF インスタンス Enterprise を同じ SNMPv2c パブリック コミュニティ ストリングにマッピングする例を示します。

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# feature ospf
switch(config)# router ospf Enterprise
switch(config-router)# exit
switch(config)# snmp-server context public1 instance Enterprise
switch(config)# snmp-server mib community-map public context public1
```

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
IP ACL と AAA	『*Cisco Nexus 9000 Series NX-OS Security Configuration Guide*』
MIB	『*Cisco Nexus 7000 Series and 9000 Series NX-OS MIB Quick Reference*』

RFC

RFC	タイトル
RFC 3414	シンプル ネットワーク管理プロトコル（*SNMPv3*）バージョン *3* 向けユーザベース セキュリティ モデル（*USM*）
RFC 3415	『*View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)*』

MIB

MIB	MIB のリンク
SNMP に関連する MIB	サポートされている MIB を検索およびダウンロ 次の URL にアクセスしてください。 https://cisco.github.io/cisco-mibs/supportlists/nexus9 Nexus9000MIBSupportList.html

翻訳について
このドキュメントは、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。