CHAPTER 10

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。

ここでは、次の内容を説明します。

- 「SNMP の概要」（P.10-1）
- 「SNMP のライセンス要件」（P.10-6）
- 「SNMP の前提条件」（P.10-7）
- 「注意事項および制約事項」（P.10-7）
- 「デフォルト設定」（P.10-7）
- 「SNMP の設定」（P.10-7）
- 「SNMP コンフィギュレーションの確認」（P.10-28）
- 「SNMP のコンフィギュレーション例」（P.10-28）
- 「その他の関連資料」（P.10-29）
- 「SNMP 機能の履歴」（P.10-30）

SNMP の概要

Simple Network Management Protocol（SNMP; 簡易ネットワーク管理プロトコル）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。

ここでは、次の内容について説明します。

- 「SNMP 機能の概要」（P.10-2）
- 「SNMP 通知」（P.10-2）
- 「SNMPv3」（P.10-2）
- 「SNMP および EEM」（P.10-5）
- 「マルチインスタンス サポート」（P.10-5）
- 「ハイ アベイラビリティ」（P.10-6）
- 「仮想化サポート」（P.10-6）

SNMP 機能の概要

SNMP フレームワークは、3 つの部分からなります。

- SNMP マネージャ：SNMP を使用してネットワーク デバイスの動作を制御およびモニタするためのシステム。
- SNMP エージェント：管理デバイス内部のソフトウェア コンポーネント。デバイスに関するデータを維持し、必要に応じてこれらのデータを管理システムに伝えます。Cisco NX-OS 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間の関係を定義する必要があります。
- Management Information Base（MIB; 管理情報ベース）：SNMP エージェント上の管理対象オブジェクトのコレクション。

SNMP は RFC 3411 ～ 3418 で定義されています。

Cisco NX-OS 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のセキュリティ形式を使用します。

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作成できるということです。これらの通知は、SNMP マネージャからの要求送信を必要としません。通知によって、不正なユーザ認証、再起動、接続の終了、ネイバー ルータとの接続切断、またはその他の重要イベントを示すことができます。

Cisco NX-OS はトラップまたは応答要求のどちらかとして 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（「VRF を使用する SNMP 通知レシーバの設定」（P.10-13）を参照）。応答要求は、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ことを確認応答で伝える必要があります。

トラップは応答要求より信頼性が低くなります。トラップの受信時に、SNMP マネージャが確認応答を送信しないので、Cisco NX-OS はトラップが受信されたかどうかを判断できないからです。応答要求を受信した場合、SNMP マネージャは SNMP 応答 Protocol Data Unit（PDU; プロトコル データ ユニット）を使用して、メッセージを確認します。応答がなかった場合、Cisco NX-OS はもう一度、応答要求を送信します。

複数のホスト レシーバに通知を送信するように、Cisco NX-OS を設定できます。ホスト レシーバの詳細については、「SNMP 通知レシーバの設定」（P.10-11）を参照してください。

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：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：パケット内容のスクランブルによって、不正な送信元で判読できないようにします。

SNMPv3 は、セキュリティ モデルとセキュリティ レベルの両方を提供します。セキュリティ モデルは、ユーザおよびユーザに与えられている役割に合わせて設定される認証方式です。セキュリティ レベルは、セキュリティ モデル内で許可されるセキュリティ レベルです。セキュリティ モデルとセキュリティ レベルのコンビネーションによって、SNMP パケットを取り扱うときに使用するセキュリティ メカニズムが決まります。

ここでは、次の内容について説明します。

- 「SNMPv1、v2、v3 のセキュリティ モデルおよびセキュリティ レベル」（P.10-3）
- 「ユーザベースのセキュリティ モデル」（P.10-4）
- 「CLI および SNMP ユーザの同期」（P.10-4）
- 「グループベースの SNMP アクセス」（P.10-5）

SNMPv1、v2、v3 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 レベルによって、SNMP メッセージを開示から保護する必要があるか、メッセージの認証が必要かどうかが決定されます。セキュリティ モデル内に存在する各種セキュリティ レベルは、次のとおりです。

- noAuthNoPriv：認証も暗号化も行わないセキュリティ レベル
- authNoPriv：認証は行うが暗号化は行わないセキュリティ レベル
- authPriv：認証と暗号化の両方を行うセキュリティ レベル

SNMPv1、SNMPv2c、SNMPv3 という 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 レベルと組み合わされたセキュリティ モデルによって、SNMP メッセージの処理時に適用されるセキュリティ メカニズムが決まります。

表 10-1 に、セキュリティ モデルとセキュリティ レベルのコンビネーションの意味を示します。

表 10-1　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	動作
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	noAuthNoPriv	ユーザ名	なし	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	なし	Hash-Based Message Authentication Code（HMAC）Message Digest 5（MD5）アルゴリズムまたは HMAC Secure Hash Algorithm（SHA）に基づいて認証します。
v3	authPriv	HMAC-MD5 または HMAC-SHA	DES	HMAC-MD5 または HMAC-SHA アルゴリズムに基づいて認証します。Cipher Block Chaining（CBC; 暗号ブロック連鎖）Data Encryption Standard（DES ; データ暗号規格）56 規格に基づいた認証に加え、DES 56 ビット暗号化を行います。

ユーザベースのセキュリティ モデル

SNMPv3 User-Based Security Model（USM; ユーザベース セキュリティ モデル）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。
- メッセージ起点認証：ユーザのために受信したデータの起点として主張されたアイデンティティが確認されていることを保証します。
- メッセージの機密性：不正な個人、エンティティ、またはプロセスに対して、情報が使用可能になったり開示されたりしていないことを保証します。

SNMPv3 で管理操作が許可されるのは、設定ユーザおよび暗号化 SNMP メッセージによる場合だけです。

Cisco NX-OS では、SNMPv3 に対応する 2 種類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

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; 高度暗号化規格）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に DES を使用するか、それとも 128 ビット AES 暗号化を使用するかを選択できます。**priv** オプションと **aes-128** トークンを組み合わせた場合は、このプライバシー パスワードが 128 ビットの AES キーを作成するためのものであることを意味します。AES priv パスワードは、8 文字以上の長さにできます。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。ローカライズしたキーを使用する場合は、130 文字まで指定できます。

（注）　外部 AAA サーバを使用する SNMPv3 動作の場合は、外部 AAA サーバ上のユーザ コンフィギュレーションで、プライバシー プロトコルとして AES を使用する必要があります。

CLI および SNMP ユーザの同期

SNMPv3 のユーザ管理は、Access Authentication and Accounting（AAA; 認証、認可、アカウンティング）サーバ レベルで集中させることができます。この集中ユーザ管理によって、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活用できます。ユーザ認証が確認されると、SNMP PDU がさらに処理されます。また、ユーザ グループ名の保管に AAA サーバも使用されます。SNMP で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およびロール 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コンフィギュレーションを変更すると、SNMP と AAA の両方について、データベースの同期が図られます。

Cisco NX-OS では次のように、ユーザ設定を同期させます。

- **snmp-server user** コマンドで指定された認証パスフレーズが 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（役割）間のマッピング変更は、SNMP と CLI で同期します。
- 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。

（注）　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 コンフィギュレーションを 60 分間維持します。このデフォルト値の変更方法については、「AAA 同期時間の変更」（P.10-28）を参照してください。

グループベースの SNMP アクセス

（注）　*グループ*が業界全体で使用されている標準的な 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 のアクセス権は、グループ別に編成されます。SNMP の各グループは、CLI でのロールと同様です。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。

自分のユーザ名を作成すると、エージェントとの通信を開始し、管理者に自分のロールを設定してもらい、そのロールに自分を追加してもらうことができます。

SNMP および EEM

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。

EEM の詳細については、第 13 章「Embedded Event Manager の設定」を参照してください。

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、VRF など、論理ネットワーク エンティティのインスタンスを複数サポートできます。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 コンテキストを論理ネットワーク エンティティにマッピングする手順は、次のとおりです。

ステップ 1　SNMPv3 コンテキストを作成します。

ステップ 2　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。

ステップ 3　SNMPv3 コンテキストを論理ネットワーク エンティティにマッピングします。

ステップ 4　任意で、SNMPv3 コンテキストを SNMPv2c コミュニティにマッピングします。

詳細については、「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」（P.10-25）を参照してください。

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 スイッチオーバー後に、Cisco NX-OS は実行コンフィギュレーションを適用します。

仮想化サポート

Cisco NX-OS は、Virtual Device Context（VDC; 仮想デバイス コンテキスト）ごとに SNMP インスタンスを 1 つずつサポートします。デフォルトでは、Cisco NX-OS はデフォルトの VDC が使用されるようにします。詳細については、『*Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 5.x*』を参照してください。

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。詳細については、「マルチインスタンス サポート」（P.10-5）を参照してください。

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。詳細については、「VRF を使用する SNMP 通知レシーバの設定」（P.10-13）を参照してください。

SNMP のライセンス要件

製品	ライセンス要件
Cisco NX-OS	SNMP にはライセンスは不要です。ライセンス パッケージに含まれていない機能は、Cisco NX-OS システム イメージにバンドルされて提供されます。追加料金は発生しません。Cisco NX-OS ライセンス方式の詳細については、『*Cisco NX-OS Licensing Guide*』を参照してください。

SNMP の前提条件

SNMP の前提条件は、次のとおりです。

- VDC を設定する場合は、Advanced Services ライセンスをインストールし、所定の VDC を開始する必要があります。詳細については、『*Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 5.x*』を参照してください。

注意事項および制約事項

SNMP に関する設定時の注意事項および制約事項は、次のとおりです。

- Cisco NX-OS は一部の SNMP MIB について、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。

  http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

デフォルト設定

表 10-2 に、SNMP パラメータのデフォルト設定を示します。

表 10-2　デフォルトの SNMP パラメータ

パラメータ	デフォルト
ライセンス通知	イネーブル

SNMP の設定

ここでは、次の内容について説明します。

- 「SNMP ユーザの設定」（P.10-8）
- 「SNMP メッセージ暗号化の強制」（P.10-9）
- 「複数のロールに SNMPv3 ユーザを割り当てる場合」（P.10-10）
- 「SNMP コミュニティの作成」（P.10-10）
- 「SNMP 要求のフィルタリング」（P.10-10）
- 「SNMP 通知レシーバの設定」（P.10-11）
- 「SNMP 通知用の発信元 インターフェイスの設定」（P.10-12）
- 「通知ターゲット ユーザの設定」（P.10-12）
- 「VRF を使用する SNMP 通知レシーバの設定」（P.10-13）
- 「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」（P.10-14）
- 「SNMP 通知のイネーブル」（P.10-16）
- 「インターフェイスに関する linkUp/linkDown 通知のディセーブル」（P.10-23）
- 「インターフェイスの SNMP ifIndex の表示」（P.10-23）
- 「TCP による SNMP のワンタイム認証のイネーブル」（P.10-24）

- 「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」（P.10-24）
- 「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」（P.10-25）
- 「SNMP のディセーブル化」（P.10-27）
- 「AAA 同期時間の変更」（P.10-28）

（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。

SNMP ユーザの設定

SNMP ユーザを設定できます。

操作の前に

正しい VDC を使用していることを確認します。VDC の変更は **switchto vdc** コマンドを使用します。

手順の概要

1. **config t**
2. **snmp-server user** *name* [**auth** {**md5** | **sha**} *passphrase* [**auto**] [**priv** [**aes-128**] *passphrase*] [**engineID** *id*] [**localizedkey**]]
3. **show snmp user**
4. **copy running-config startup-config**

手順の詳細

	コマンド	目的
ステップ 1	**config t** 例： switch# config t Enter configuration commands, one per line. End with CNTL/Z. 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**} *passphrase* [auto] [**priv** [aes-128] *passphrase*] [engineID *id*] [localizedkey]] 例： switch(config)# snmp-server user Admin auth sha abcd1234 priv abcdefgh	認証およびプライバシー パラメータを指定して、SNMP ユーザを設定します。パスフレーズには最大 64 の英数字を使用できます。大文字と小文字を区別します。**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。 engineID の形式は、12 桁のコロンで区切った 10 進数字です。

	コマンド	目的
ステップ 3	**show snmp user** **例：** switch(config-callhome)# show snmp user	（任意）1 つまたは複数の SNMP ユーザに関する情報を表示します。
ステップ 4	**copy running-config startup-config** **例：** switch(config)# copy running-config startup-config	（任意）このコンフィギュレーションの変更を保存します。

SNMP のコンタクトおよびロケーション情報を設定する例を示します。

```
switch# config t
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# snmp-server user Admin auth sha abcd1234 priv abcdefgh
```

SNMP メッセージ暗号化の強制

着信要求の認証または暗号化を求めるように、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ずに SNMPv3 メッセージを受け付けます。プライバシーを強化する場合、Cisco NX-OS は noAuthoNoPriv または authNoPriv の securityLevel パラメータを使用している SNMPv3 PDU に、authorizationError で応答します。

SNMP メッセージの暗号化をユーザに強制するには、グローバル 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 user *name* enforcePriv **例：** switch(config)# snmp-server user Admin enforcePriv	このユーザに SNMP メッセージの暗号化を強制します。

SNMP メッセージの暗号化をすべてのユーザに強制するには、グローバル 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 globalEnforcePriv **例：** switch(config)# snmp-server globalEnforcePriv	すべてのユーザに SNMP メッセージの暗号化を強制します。

複数のロールに SNMPv3 ユーザを割り当てる場合

SNMP ユーザの設定後、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 ロールに属しているユーザだけです。

SNMP ユーザにロールを割り当てるには、グローバル 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 user *name group* **例：** `switch(config)# snmp-server user Admin superuser`	この SNMP ユーザを設定済みのユーザ ルールに関連付けます。

SNMP コミュニティの作成

SNMPv1 または SNMPv2c に対応する SNMP コミュニティを作成できます。

SNMP コミュニティ ストリングを作成するには、グローバル 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 community *name 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。

SNMP 要求のフィルタリング

コミュニティにアクセス リスト（ACL）を割り当てると、SNMP の 着信要求をフィルタできます。割り当てられた ACL で着信要求パケットが許可されている場合、SNMP は要求を処理します。ACL が要求を拒否する場合、SNMP は要求をドロップし、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。

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

ACL 作成の詳細については、『*Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 5.x*』を参照してください。ACL は、UDP および TCP 経由の IPv4 および IPv6 の両方に適用されます。

ACL をコミュニティに割り当てて SNMP 要求をフィルタするには、グローバル 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 community *community-name* **use-acl** *acl-name* **例：** `switch(config)# snmp-server community public use-acl my_acl_for_public`	ACL を SNMP コミュニティに割り当てて SNMP 要求をフィルタします。

SNMP 通知レシーバの設定

複数のホスト レシーバに対して SNMP 通知を作成するように、Cisco NX-OS を設定できます。

SNMPv1 トラップのホスト レシーバを設定するには、グローバル 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 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。*community* 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。

SNMPv2c トラップまたは応答要求のホスト レシーバを設定するには、グローバル 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 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。*community* 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。

SNMPv3 トラップまたは応答要求のホスト レシーバを設定するには、グローバル 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 host *ip-address* {**traps** | **informs**} **version 3** {**auth** | **noauth** | **priv**} *username* [**udp_port** *number*] **例：** `switch(config)# snmp-server host 192.0.2.1 informs version 3 auth NMS`	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。*username* 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。

（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 エンジン ID 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。

SNMP 通知用の発信元 インターフェイスの設定

特定のインターフェイスを通知用の発信元 インターフェイスとして使用するよう、SNMP を設定できます。これは、次のように設定できます。

- 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。

発信元インターフェイスでホスト レシーバを設定するには、グローバル 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 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。サポートされているインターフェイス タイプを調べるには、**?** 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。

すべての SNMP 通知を送信するよう発信元インターフェイスを設定するには、グローバル 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 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 タイプを調べるには、**?** を使用します。

設定されている発信元インターフェイスの情報を表示するには、**show snmp source-interface** コマンドを使用します。

通知ターゲット ユーザの設定

通知ホスト レシーバに SNMPv3 応答要求通知を送信するには、デバイス上で通知ターゲット 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）　受信した応答要求 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。

通知ターゲット ユーザを設定するには、グローバル 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 user *name* [auth {md5 | sha} *passphrase* [auto] [**p**riv [aes-128] *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。engineID の形式は、12 桁のコロンで区切った 10 進数字です。

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。

設定された VRF を使用してホスト レシーバに接続するように Cisco NX-OS を設定できます。

ホスト レシーバへの通知の送信に使用する VRF を設定するには、グローバル 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 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* 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。
no snmp-server host *ip-address* **use-vrf** *vrf_name* [**udp_port** *number*] **例：** `switch(config)# no snmp-server host 192.0.2.1 use-vrf Blue`	設定済みホストの 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。 *ip-address* は IPv4 または IPv6 アドレスを使用できます。 ホスト設定は削除されません。

通知が発生した VRF に基づいて、通知をフィルタリングするように Cisco NX-OS を設定できます。

設定された VRF に基づいて通知をフィルタリングするには、グローバル 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 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。
no snmp-server host *ip-address* **filter-vrf** *vrf_name* **例：** `switch(config)# no snmp-server host 192.0.2.1 filter-vrf Red`	設定済みホストの 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。 *ip-address* は IPv4 または IPv6 アドレスを使用できます。このコマンドによってホスト設定は削除されません。

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. **config t**
2. **snmp-server source-interface traps** *if-type if-number*
3. **show snmp source-interface**
4. **snmp-server host** *ip-address* **use-vrf** *vrf_name* [**udp_port** ***number***]
5. **show snmp host**
6. **copy running-config startup-config**

手順の詳細

	コマンド	目的

ステップ 1	**config t**
	
	例：
	`switch# config t`
	`Enter configuration commands, one per line.  End with CNTL/Z.`
	`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 タイプを調べるには、**?** 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	**show snmp source-interface** **例：** `switch(config)# show snmp source-interface`	（任意）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	**show snmp host** **例：** `switch(config)# show snmp host`	（任意）設定した SNMP ホストの情報を表示します。
ステップ 6	**copy running-config startup-config** **例：** `switch(config)# copy running-config startup-config`	（任意）このコンフィギュレーションの変更を保存します。

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# config t
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
-----------------------------------------------------------------------
```

SNMP 通知のイネーブル

通知をイネーブルまたはディセーブルにできます。通知の名前を指定しなかった場合、Cisco NX-OS はすべての通知をイネーブルにします。

表 10-3 に、Cisco NX-OS MIB に関する通知をイネーブルにする、コマンドを示します。

（注）　**snmp-server enable traps** コマンドを使用すると、設定されている通知ホスト サーバに応じて、トラップおよび応答要求の両方がイネーブルになります。

表 10-3　SNMP 通知のイネーブル

MIB	関連コマンド
すべての通知	**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-STP-BRIDGE-MIB	**snmp-server enable traps bridge** **snmp-server enable traps bridge newroot** **snmp-server enable traps bridge topologychange**
CISCO-CALLHOME-MIB	**snmp-server enable traps callhome** **snmp-server enable traps callhome event-notify** **snmp-server enable traps callhome smtp-send-fail**

表 10-3　SNMP 通知のイネーブル（続き）

MIB	関連コマンド
CISCO-CFS-MIB	**snmp-server enable traps cfs** **snmp-server enable traps cfs merge-failure** **snmp-server enable traps cfs state-change-notif**
CISCO-CONFIG-MAN-MIB	**snmp-server enable traps config** **snmp-server enable traps config ccmCLIRunningConfigChanged**
CISCO-EIGRP-MIB	**snmp-server enable traps eigrp** [*tag*]
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**
CISCO-LICENSE-MGR-MIB	**snmp-server enable traps license** **snmp-server enable traps license notify-license-expiry** **snmp-server enable traps license notify-license-expiry-warning** **snmp-server enable traps license notify-licensefile-missing** **snmp-server enable traps license notify-no-license-for-feature**
IF-MIB	**snmp-server enable traps link** **snmp-server enable traps link IETF-extended-linkDown** **snmp-server enable traps link IETF-extended-linkUp** **snmp-server enable traps link cisco-extended-linkDown** **snmp-server enable traps link cisco-extended-linkUp** **snmp-server enable traps link linkDown** **snmp-server enable traps link Up**
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*
CISCO-PORT-SECURITY-MIB	**snmp-server enable traps port-security** **snmp-server enable traps port-security access-secure-mac-violation** **snmp-server enable traps port-security trunk-secure-mac-violation**
CISCO-RF-MIB	**snmp-server enable traps rf** **snmp-server enable traps rf redundancy_framework**

表 10-3　SNMP 通知のイネーブル（続き）

MIB	関連コマンド
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-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	**sysmgr** **sysmgr cseFailSwCoreNotifyExtended**
UPGRADE-MIB	**upgrade** **upgrade UpgradeJobStatusNotify** **upgrade UpgradeOpNotifyOnCompletion**
ZONE-MIB	**zone** **zone default-zone-behavior-change** **zone merge-failure** **zone merge-success** **zone request-reject1** **zone unsupp-mem**

ライセンス通知は、デフォルトでイネーブルです。その他の通知はすべて、デフォルトでディセーブルです。

指定した通知をイネーブルにするには、グローバル 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イネーブルにします。
snmp-server enable traps bgp **例：** `switch(config)# snmp-server enable traps bgp`	BGP SNMP 通知をイネーブルにします。

コマンド	目的
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 ブリッジ通知をイネーブルにします。 • **topologychange**：STP ブリッジのトポロジ変更通知をイネーブルにします。
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イネーブルにします。 • **smtp-send-fail**：Simple Mail Transfer Protocol（SMTP; 簡易メール転送プロトコル）メッセージの送信失敗通知をイネーブルにします。
snmp-server enable traps cfs [**merge-failure**] [**state-change-notif**] **例：** `switch(config)# snmp-server enable traps cfs`	Cisco Fabric Services（CFS）の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。 • **merge-failure**：CFS のマージ失敗通知をイネーブルにします。 • **state-change-notif**：CFS の状態変化通知をイネーブルにします。
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 通知をイネーブルにします。
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。

コマンド	目的
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イネーブルにします。 • **entity_mib_change**：エンティティ MIB 変更通知をイネーブルにします。 • **entity_module_inserted**：エンティティ モジュール挿入通知をイネーブルにします。 • **entity_module_removed**：エンティティ モジュール削除通知をイネーブルにします。 • **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イネーブルにします。 • **entity_power_out_change**：エンティティの出力パワー変更通知をイネーブルにします。 • **entity_power_status_change**：エンティティのパワー ステータス変更通知をイネーブルにします。 • **entity_unrecognised_module**：エンティティの未確認モジュール通知をイネーブルにします。
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イネーブルにします。
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 の状態変化通知をイネーブルにします。
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`	ライセンス 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イネーブルにします。 • **notify-license-expiry-warning**：ライセンス失効の警告通知をイネーブルにします。 • **notify-licensefile-missing**：ライセンス ファイル不明通知をイネーブルにします。 • **notify-no-license-for-feature**：no-license-installed-for-feature 通知をイネーブルにします。

コマンド	目的
snmp-server enable traps link [**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、次の特定の通知をイネーブルにします。 • **IETF-extended-linkDown**：Internet Engineering Task Force（IETF; インターネット技術特別調査委員会）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。 • **IETF-extended-linkUp**：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。 • **cisco-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。 • **cisco-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。 • **linkDown**：IETF リンク ステート ダウン通知をイネーブルにします。 • **linkUp**：IETF リンク ステート アップ通知をイネーブルにします。
snmp-server enable traps ospf [*tag*] [**lsa**] [**rate-limit** *rate*] **例：** `switch(config)# snmp-server enable traps ospf`	Open Shortest Path First（OSPF）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。 • **lsa**：OSPF LSA 通知をイネーブルにします。 • **rate-limit** *rate*：OSPF のレート制限通知をイネーブルにします。値の範囲は 2 ～ 60 秒です。デフォルト値は 10 秒です。
snmp-server enable traps port-security [**access-secure-mac-violation**] [**trunk-secure-mac-violation**] **例：** `switch(config)# snmp-server enable traps port-security`	ポート セキュリティ 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。 • **access-secure-mac-violation**：セキュアな Machine Access Control（MAC）違反通知をイネーブルにします。 • **trunk-secure-mac-violation**：仮想 LAN（VLAN）のセキュア MAC 違反通知をイネーブルにします。
snmp-server enable traps rf [**redundancy-framework**] **例：** `switch(config)# snmp-server enable traps rf`	Redundancy Framework（RF; 冗長フレームワーク）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 通知をイネーブルにします。

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イネーブルにします。 • **hcFallingAlarm**：RMON high-capacity 下限アラーム通知をイネーブルにします。 • **hcRisingAlarm**：RMON high-capacity 上限アラーム通知をイネーブルにします。 • **risingAlarm**：RMON 上限アラーム通知をイネーブルにします。
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 認証通知をイネーブルにします。
snmp-server enable traps stpx [**inconsistency**] [**loop-inconsistency**] [**root-inconsistency**] **例：** `switch(config)# snmp-server enable traps stpx`	STPX MIB 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。 • **inconsistency**：SNMP STPX MIB 不一致アップデート通知をイネーブルにします。 • **loop-inconsistency**：SNMP STPX MIB ループ不一致アップデート通知をイネーブルにします。 • **root-inconsistency**：SNMP STPX MIB ルート不一致アップデート通知をイネーブルにします。
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 コア通知をイネーブルにします。

コマンド	目的
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 ステータス通知をイネーブルにします。 • **UpgradeOpNotifyOnCompletion**：アップグレード グローバル ステータス通知をイネーブルにします。
snmp-server enable traps zone [**default-zone-behavior-change**] [**merge-failure**] [**merge-success**] [**request-reject1**] [**unsupp-mem**] **例：** `switch(config)# snmp-server enable traps zone`	デフォルト ゾーン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。 • **default-zone-behavior-change**：デフォルト ゾーン動作変更通知をイネーブルにします。 • **merge-failure**：マージ失敗通知をイネーブルにします。 • **merge-success**：マージ成功通知をイネーブルにします。 • **request-reject1**：要求拒否通知をイネーブルにします。 • **unsupp-mem**：未サポート メンバ通知をイネーブルにします。

インターフェイスに関する linkUp/linkDown 通知のディセーブル

個々のインターフェイスに関する linkUp および linkDown 通知をディセーブルにできます。これにより、フラッピング インターフェイス（アップとダウン間の移行を繰り返しているインターフェイス）に関する通知を制限できます。

インターフェイスに関する linkUp/linkDown 通知をディセーブルにするには、インターフェイス コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
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。

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。ifIndex は、NetFlow がインターフェイスの情報を収集する際にも使用されます。

インターフェイスの SNMP ifIndex 値を表示するには、任意のモードで次のコマンドを使用します。

コマンド	目的

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。| 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。

TCP による SNMP のワンタイム認証のイネーブル

TCP セッションでの 1 回限りの SNMP 認証をイネーブルにできます。

TCP による SNMP のワンタイム認証をイネーブルにするには、グローバル 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 tcp-session [**auth**] **例：** `switch(config)# snmp-server tcp-session`	TCP セッションでの 1 回限りの SNMP 認証をイネーブルにします。デフォルトはディセーブルです。

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。

操作の前に

正しい VDC を使用していることを確認します。VDC の変更は **switchto vdc** コマンドを使用します。

手順の概要

1. **config t**
2. **snmp-server contact** *name*
3. **snmp-server location** *name*
4. **show snmp**
5. **copy running-config startup-config**

手順の詳細

	コマンド	目的

ステップ 1	**config t**
	
	例：
	`switch# config t`
	`Enter configuration commands, one per line.  End with CNTL/Z.`
	`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* 例： `switch(config)# snmp-server contact Admin`	SNMP コンタクト名として sysContact を設定します。
ステップ 3	**snmp-server location** *name* 例： `switch(config)# snmp-server location Lab-7`	SNMP ロケーションとして sysLocation を設定します。
ステップ 4	**show snmp** 例： `switch(config)# show snmp`	（任意）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	**copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	（任意）このコンフィギュレーションの変更を保存します。

SNMP のコンタクトおよびロケーション情報を設定する例を示します。

```
switch# config t
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# snmp contact Admin
switch(config)# snmp location Lab-7
```

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。

操作の前に

正しい VDC を使用していることを確認します。VDC の変更は **switchto vdc** コマンドを使用します。

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 インスタスの詳細については、『*Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide, Release 5.x*』または『*Cisco Nexus 7000 Series NX-OS Multicast Routing Configuration Guide, Release 5.x*』を参照してください。

手順の概要

1. **config t**
2. **snmp-server context** *context-name* [**instance** *instance-name*] [**vrf** *vrf-name*] [**topology** *topology-name*]
3. **snmp-server mib community-map** *community-name* **context** ***context-name***
4. **show snmp context**
5. **copy running-config startup-config**

手順の詳細

	コマンド	目的

ステップ 1	**config t**
	
	例：
	`switch# config t`
	`Enter configuration commands, one per line.  End with CNTL/Z.`
	`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 context** *context-name* [**instance** *instance-name*] [**vrf** *vrf-name*] [**topology** *topology-name*] 例： `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。
ステップ 3	**snmp-server mib community-map** *community-name* **context** *context-name* 例： `switch(config)# snmp-server mib community-map public context public1`	（任意）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	**show snmp context** 例： `switch(config)# show snmp context`	（任意）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。
ステップ 5	**copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	（任意）このコンフィギュレーションの変更を保存します。

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# config t
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# config t
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
```

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を削除するには、グローバル 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
no snmp-server context *context-name* [**instance** *instance-name*] [**vrf** *vrf-name*] [**topology** *topology-name*] **例：** `switch(config)# no snmp-server context public1`	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。名前には最大 32 の英数字を使用できます。 **（注）** 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。**instance**、**vrf**、または **topology** 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。

SNMP のディセーブル化

デバイス上で SNMP をディセーブルにすることができます。

SNMP をディセーブルにするには、グローバル 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
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。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。

AAA 同期時間の変更

同期したユーザ コンフィギュレーションを Cisco NX-OS に維持させる時間の長さを変更できます。

AAA 同期時間を変更するには、グローバル 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 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 コンフィギュレーションを維持する時間を設定します。値の範囲は 1 ～ 86400 秒です。デフォルト値は 3600 です。

SNMP コンフィギュレーションの確認

SNMP のコンフィギュレーション情報を表示するには、次の作業のいずれかを行います。

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。
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。
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。
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。
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	SNMP 通知がイネーブルなのかディセーブルなのかを表示します。
show snmp user	SNMPv3 ユーザを表示します。

SNMP のコンフィギュレーション例

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。

```
config t
 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# config t
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
-------------------------------------------------------------------
```

その他の関連資料

SNMP の実装に関連する詳細情報については、次の項を参照してください。

- 「関連資料」（P.10-29）
- 「規格」（P.10-30）
- 「MIB」（P.10-30）

関連資料

関連項目	マニュアル名
SNMP CLI コマンド	『*Cisco Nexus 7000 Series NX-OS System Management Command Reference, Release 5.x*』
VDC および VRF	『*Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 5.x*』
MIB	http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

規格

規格	タイトル
この機能がサポートする新しい規格または変更された規格はありません。また、この機能で変更された既存規格のサポートはありません。	—

MIB

MIB	MIB のリンク
• SNMP-COMMUNITY-MIB • SNMP-FRAMEWORK-MIB • SNMP-NOTIFICATION-MIB • SNMP-TARGET-MIB • SNMPv2-MIB	MIB を見つけてダウンロードするには、次の URL を参照してください。 http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

SNMP 機能の履歴

表 10-4 に、この機能のリリース履歴を示します。

表 10-4　SNMP 機能の履歴

機能名	リリース	機能情報
SNMP 通知	5.0(2)	**snmp-server enable traps** コマンドが更新されました。「SNMP 通知のイネーブル」（P.10-16）を参照してください。
IPv6 サポート	4.2(1)	IPv6 SNMP ホストの設定がサポートされました。
ACL を使用したコミュニティでの SNMP 要求フィルタ	4.2(1)	ACL を SNMP コミュニティに割り当てて SNMP 要求をフィルタします。「SNMP 要求のフィルタリング」（P.10-10）を参照してください。
SNMP 通知レシーバ用インターフェイスの使用	4.2(1)	インターフェイスを SNMP 通知用の発信元インターフェイスとして機能するよう指定する機能のサポートが追加されました。「SNMP 通知レシーバの設定」（P.10-11）を参照してください。
SNMP AAA 同期	4.0(3)	同期したユーザ設定のタイムアウトを変更する機能が追加されました。「AAA 同期時間の変更」（P.10-28）を参照してください。
SNMP プロトコル	4.0(3)	SNMP プロトコルをディセーブルにする機能を追加「SNMP のディセーブル化」（P.10-27）を参照してください。