# Flexible NetFlow の設定



## 機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。最新の機能情報および警告については、「Bug Search Tool」およびご使用のプラットフォームおよびソフトウェア リリースのリリース ノートを参照してください。このモジュールで説明される機能に関する情報、および各機能がサポートされるリリースの一覧については、機能情報の表を参照してください。

プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

## NetFlow Lite の前提条件

NetFlow Lite モニターを接続するために、次の 2 つのターゲットがサポートされています。

- ポート：EtherChannel などの論理インターフェイスではなく、物理インターフェイスのみでサポートされるモニター接続。物理インターフェイスは、ルーテッド ポートまたはスイッチド ポートです。
- VLAN：モニター接続は、レイヤ 2 VLAN ではなく、VLAN インターフェイス（SVI）のみでサポートされます。

# NetFlow Lite の制約事項

NetFlow Liteの制限事項は次のとおりです。

- モニタの制約事項：
  - モニター接続は、入力方向に限りサポートされます。
  - エクスポータはインターフェイスごとに複数サポートされますが、モニターはインターフェイスごとに 1 つサポートされます。
  - モニターでは永続キャッシュと標準キャッシュのみサポートされます。即時キャッシュはサポートされません。
  - モニター パラメータがインターフェイスまたは VLAN に適用される場合は、それらのモニター パラメータは変更できません。
  - ポートおよびVLANの両方がモニターに接続されている場合、ポートに着信するトラフィックについて VLAN モニターはポート モニターを上書きします。
  - フローモニタータイプとトラフィック タイプ（タイプとは、IPv4、IPv6、およびデータ リンクを意味します）は、作成するフローで同じである必要があります。
  - deviceでは、インターフェイスにIPおよびポートベースのモニターを同時に接続できません。48 ポートdeviceは最大 48 台のモニター（IP またはポートベース）をサポートし、256 SVI は最大 256 台のモニター（IP またはポートベース）を設定できます。
  - **show flow monitor** *flow_name***cache** コマンドを実行すると、スイッチはそれ以前のスイッチ ソフトウェア バージョン（Catalyst 2960-S）からのキャッシュ情報を、すべてのフィールドにゼロが入力された状態で表示します。これらのフィールドはスイッチに適用できないため、無視します。
- サンプラーの制限事項：
  - サンプリングされた NetFlow のみがサポートされます。
  - ポートと VLAN の両方について、deviceでは合計 4 つのサンプラーのみ（ランダムまたは確定）がサポートされます。
  - 両方のモードのサンプリング最小レートは、32 個のフローの中から 1 つで、両方のモードのサンプリング最大レートは 1022 個のフローから 1 つです。
  - サンプラーをインターフェイスに接続している間、サンプラーをモニタと関連付けておく必要があります。これを行わないと、コマンドは拒否されます。このタスクを実行するには、**ip flow monitor** *monitor_name* sampler s*ampler_name* **input** インターフェイス コンフィギュレーション コマンドを使用します。
  - 確定サンプラーを使用してモニタを接続する場合は、同じサンプラーを使用するすべての接続で、4 個の使用可能なサンプラーの中から 1 つの新しいフリー サンプラーを

スイッチ（ハードウェア）から使用します。サンプラーによるモニターの接続は4つまで許容されます。

ランダムサンプラーを使用してモニターを接続する場合は、最初の接続のみがスイッチ（ハードウェア）からの新しいサンプラーを使用します。同じサンプラーを使用する残りのすべての接続は、同じサンプラーを共有します。

この動作のため、確定サンプラーを使用する場合は、サンプリングレートとdeviceが送信した内容を比較することによって、サンプリングされたフローの正確な数を常に確認できます。同じランダム サンプラーを複数のインターフェイスで使用する場合は、任意のインターフェイスからのフローを常にサンプリングし、他のインターフェイスからのフローは常にスキップすることができます。

- ネットワーク フローおよび統計情報はライン レートで収集されます。
- ACL ベースの NetFlow はサポートされていません。
- NetFlow バージョン 9 のみが *export-protocol* コマンド オプションを使用した Flexible NetFlow エクスポータでサポートされます。NetFlow バージョン 5 を設定した場合、このバージョンは受け入れられますが、現在、NetFlow バージョン 5 のエクスポート機能は利用できず、サポートもされていません。
- スイッチは同種スタック構成をサポートしますが、混合スタック構成はサポートしません。

# NetFlow Lite について

## NetFlow Lite の概要

NetFlow Lite ではフローを使用して、アカウンティング、ネットワーク モニターリング、およびネットワーク プランニングに関連する統計情報を提供します。

フローは送信元インターフェイスに届く単方向のパケット ストリームで、キーの値は同じです。キーは、パケット内のフィールドを識別する値です。フローを作成するには、フロー レコードを使用して、フロー固有のキーを定義します。

deviceは、ネットワーク異常とセキュリティ問題の高度な検出をイネーブルにする NetFlow Lite 機能をサポートします。NetFlow Lite により、大量の定義済みフィールドの集合からキーを選択して、特定のアプリケーションに最適なフロー レコードを定義できます。

1 つのフローと見なされるパケットでは、すべてのキー値が一致している必要があります。フローは、設定したエクスポート レコード バージョンに基づいて、関係のある他のフィールドを集めることもあります。フローは NetFlow Lite キャッシュに格納されます。

エクスポータを使用してNetFlow Liteがフローのために収集するデータをエクスポートし、NetFlow Lite コレクタなどのリモート システムにこのデータをエクスポートできます。NetFlow Lite コレクタは、IPv4 アドレスを使用できます。

モニターを使用してフローのために収集するデータのサイズを定義します。モニターで、フロー レコードおよびエクスポータを NetFlow Lite キャッシュ情報と結合します。

Cisco IOS XE 16.12.1 リリース以降、Flexible NetFlow 上の送信元グループタグ（SGT）および宛先グループタグ（DGT）フィールドは、IPv6 トラフィックでサポートされます。

# Flexible NetFlow のコンポーネント

Flexible NetFlow は、いくつかのバリエーションで一緒に使用して、トラフィック分析およびデータ エクスポートに使用できるコンポーネントで構成されます。Flexible NetFlow のユーザー定義のフロー レコードおよびコンポーネントの構造では、最小限の数のコンフィギュレーション コマンドで、ネットワーキング デバイスでのトラフィック分析およびデータ エクスポートのためのさまざまなコンフィギュレーションの作成が容易になります。各フローモニターに、フロー レコード、フロー エクスポータ、およびキャッシュ タイプの固有の組み合わせを設定できます。フロー エクスポータの宛先 IP アドレスなどのパラメータを変更する場合、フロー エクスポータを使用するすべてのフローモニターに対して自動的に変更されます。同じフローモニターを複数のフローサンプラと組み合わると、さまざまなインターフェイス上でさまざな速度の同じタイプのネットワーク トラフィックをサンプリングできます。ここでは、Flexible NetFlow コンポーネントのその他の情報を提供します。

## フロー レコード

Flexible NetFlow では、キー フィールドと非キー フィールドの組み合わせをレコードと呼びます。Flexible NetFlow のレコードは Flexible NetFlow フロー モニターに割り当てられ、フローデータの格納に使用されるキャッシュが定義されます。

フロー レコードでは、フロー内のパケットを識別するために Flexible NetFlow で使用するキーとともに、Flexible NetFlow がフローについて収集する他の関連フィールドを定義します。キーと関連フィールドを任意の組み合わせで指定して、フロー レコードを定義できます。device は、幅広いキーセットをサポートします。フローレコードでは、フロー単位で収集するカウンタのタイプも定義します。64 ビットのパケットまたはバイトカウンタを設定できます。device は、フロー レコードの作成時に、デフォルトとして次の match フィールドをイネーブルにします。

- match datalink：レイヤ 2 属性
- match ipv4：IPv4 属性
- match ipv6：IPv6 属性
- match transport：トランスポート層フィールド
- match wireless：ワイヤレス フィールド

### NetFlow の事前定義済みのレコード

Flexible NetFlow には事前定義済みのレコードがいくつか含まれ、それを使用してネットワーク トラフィックの監視を開始できます。事前定義済みのレコードは、Flexible NetFlow を迅速に導入するために役立ち、ユーザー定義のフロー レコードよりも簡単に使用できます。ネット

ワーク モニタリングのニーズを満たす定義済みのレコードのリストから選択できます。Flexible NetFlowが改良されると、一般的なユーザー定義のフロー レコードを事前定義済みレコードとして使用でき、簡単に導入できるようになります。

事前定義済みレコードにより、エクスポートされるデータのために既存のNetFlow コレクタ コンフィギュレーションとの下位互換性が確保されます。事前定義済みレコードは、それぞれ固有の key および nonkey フィールドの組み合わせを持ち、ルータで Flexible NetFlow をカスタマイズしなくても、ネットワーク内のさまざまなタイプのトラフィックを監視する、内蔵機能を提供します。

2 つの事前定義済みレコード（NetFlow original と NetFlow IPv4/IPv6 original output）は機能的に同等で、以前の（入力）NetFlow、および以前の NetFlow の 出力 NetFlow アカウンティング機能をそれぞれエミュレートします。その他の Flexible NetFlow の事前定義済みレコードのいくつかは、以前の NetFlow で利用できる集約キャッシュ方式に基づきます。以前の NetFlow で利用できる集約キャッシュ方式に基づく Flexible NetFlow の事前定義済みレコードでは、集約を実行しません。代わりに、事前定義済みレコードによって各フローが個別に追跡されます。

## ユーザー定義レコード

Flexible NetFlow では、key および nonkey フィールドを指定し、実際の要件に合わせてデータ収集をカスタマイズすることで、Flexible NetFlow フロー モニター キャッシュ用の独自のレコードを定義できます。Flexible NetFlow フロー モニター キャッシュに対して独自のレコードを定義する場合、ユーザー定義レコードと呼ばれます。nonkeyフィールドの値は、フロー内のトラフィックに関する追加情報を提供するためにフローに追加されます。nonkeyフィールドの値の変更によって新しいフローが作成されることはありません。ほとんどの場合、nonkeyフィールドの値はフロー内の最初のパケットからのみ取得されます。FlexibleNetFlowを使用すると、nonkeyフィールドとして、フロー内のバイト数やパケット数などのカウンター値をキャプチャできます。

Flexible NetFlow では、ヘッダーおよびパケット セクションのタイプに新しいバージョン 9 エクスポート フォーマット フィールド タイプが追加されます。Flexible NetFlow は NetFlow コレクタに、対応するバージョン 9 エクスポート テンプレート フィールドで設定されたセクション サイズを通知します。ペイロード セクションには、対応する長さフィールドがあり、収集されるセクションの実際のサイズを収集するために使用できます。

## NetFlow Lite match パラメータ

フロー レコードの次のキー フィールドを照合できます。

- IPv4 または IPv6 宛先アドレス
- Datalink フィールド（送信元および宛先 MAC アドレス、ならびに MAC EtherType（ネットワーク プロトコルのタイプ））。
- アプリケーションのタイプ（ICMP、IGMP、または TCP トラフィック）を識別するトランスポート フィールドの送信元および宛先ポート。

次の表で、NetFlow Lite の match パラメータについて説明します。フロー レコードごとに、次の match パラメータを 1 つ以上設定する必要があります。

表 *1 : match* パラメータ

| コマンド | 目的 |
|---|---|
| **match datalink** {**ethertype** \| **mac** {**destination address input** \| **source address input**}} | データ リンクまたはレイヤ 2 フィールドとの一致を指定します。次のコマンド オプションが使用可能です。<br>• **ethertype**：パケットの ethertype と一致します。<br>• **mac**：入力時のパケットの送信元または宛先 MAC アドレスと一致します。<br>（注） データリンク フロー モニタがインターフェイスまたはVLANに割り当てられている場合、非IPv6または非IPv4トラフィック用のフローだけが作成されます。 |
| **match ipv4** {**destination** {**address**} \| **protocol** \| **source** {**address**} \| **tos**} | IPv4 フィールドとの一致を指定します。次のコマンド オプションが使用可能です。<br>• **destination**：IPv4 宛先アドレス ベースのフィールドと一致します。<br>• **protocol**：IPv4プロトコルと一致します。<br>• **source**：IPv4 送信元アドレス ベースのフィールドと一致します。<br>• **tos**：IPv4 タイプ オブ サービス フィールドと一致します。 |
| **match ipv6** {**destination** {**address**} \| **flow-label** \| **protocol** \| **source** {**address**} } | IPv6 フィールドとの一致を指定します。次のコマンド オプションが使用可能です。<br>• **destination**：IPv6 宛先アドレス ベースのフィールドと一致します。<br>• **flow-label**：IPv6 フローラベル フィールドと一致します。<br>• **protocol**：IPv6 ペイロード プロトコル フィールドと一致します。<br>• **source**：IPv6 送信元アドレス ベースのフィールドと一致します。 |

| コマンド | 目的 |
|---|---|
| **match transport** {**destination-port** \| **source-port**} | トランスポート層フィールドとの一致を指定します。次のコマンド オプションが使用可能です。<br>• **destination-port**：転送先ポートと一致します。<br>• **source-port**：転送元ポートと一致します。 |
| | フロー レコードのキー フィールドとしてSSIDのワイヤレス ネットワークの使用を指定します。 |

## NetFlow Lite collect パラメータ

フロー レコードの次のキー フィールドを収集できます。

- 合計バイト数、エクスポータによって送信されるまたはフローまたはパケット（exporter）、または 64 ビット カウンタのバイト数またはパケット数（long）。
- 最初のパケットの送信時間または最新（最後）のパケットが見つかった時間からのシステム稼働時間に基づくタイムスタンプ。
- 入力インターフェイスのSNMPインデックス。サービスモジュールに着信するトラフィックのインターフェイスは、スイッチの転送キャッシュに基づいています。このフィールドは、一般にデータ リンク、IPv4 および IPv6 アドレスとともに使用され、直接接続されたホストの実際のファースト ホップのインターフェイスを提供します。
  - 値 0 は、インターフェイス情報がキャッシュにないことを意味します。
  - 一部の NetFlow コレクタでは、フロー レコードにこの情報が必要です。

次の表で、NetFlow Liteの collect パラメータについて説明します。

***表 2 : collect* パラメータ**

| コマンド | 目的 |
|---|---|
| **collect counter** {**bytes** {**long** \| **permanent** } \| **packets** { **long** \| **permanent**}} | カウンタ フィールドの合計バイト数と合計パケット数を収集します。 |
| **collect flow** {**sampler**} | フロー サンプラー識別子（ID）を収集します。 |
| **collect interface** {**input**} | 入力インターフェイスからフィールドを収集します。 |

| コマンド | 目的 |
|---|---|
| **collect timestamp sys-uptime** {**first** \| **last**} | 最初のパケットが確認された時刻、または最新のパケットが最後に確認された時刻のフィールドを収集します（ミリ秒）。 |
| **collect transport tcp flags** | 次の転送 TCP フラグを収集します。<br>• **ack**：TCP 確認応答フラグ<br>• **cwr**：TCP 輻輳ウィンドウ縮小フラグ<br>• **ece**：TCP ECN エコー フラグ<br>• **fin**：TCP 終了フラグ<br>• **psh**：TCP プッシュ フラグ<br>• **rst**：TCP リセット フラグ<br>• **syn**：TCP 同期フラグ<br>• **urg**：TCP 緊急フラグ |
| | ワイヤレス クライアントが関連付けられているアクセス ポイントの MAC アドレスを収集します。 |

## フロー エクスポータ

フローエクスポータでは、フロー モニタ キャッシュ内のデータをリモートシステム（たとえば、分析および保管のためにNetFlowコレクタを実行するサーバ）にエクスポートします。フローエクスポータは、コンフィギュレーションで別のエンティティとして作成されます。フローエクスポータは、フローモニタにデータエクスポート機能を提供するためにフローモニタに割り当てられます。複数のフローエクスポータを作成して、1 つまたは複数のフローモニタに適用すると、いくつかのエクスポート先を指定することができます。1つのフローエクスポータを作成し、いくつかのフローモニタに適用することができます。

### NetFlow データ エクスポート フォーマットのバージョン 9

NetFlow の基本出力はフロー レコードです。NetFlow が改良され、フロー レコードのいくつかのフォーマットが向上しました。NetFlow エクスポート フォーマットの最新の進化は、バージョン 9 と呼ばれます。NetFlow Version 9 エクスポート フォーマットの識別機能は、テンプレートがベースとなります。テンプレートは、レコードフォーマットの設計を拡張可能なものにします。NetFlowサービスが将来拡張されても、基本フロー レコードフォーマットを変更し続ける必要がありません。テンプレートを使用すると、次のいくつかの利点があります。

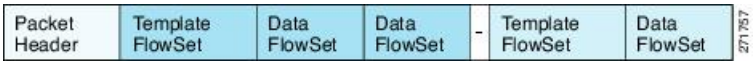
- NetFlow のコレクタを提供したり、サービスを表示したりするアプリケーションを作成するサードパーティ ビジネス パートナーは、新規の NetFlow 機能が追加されるたびにアプ

リケーションを再コンパイルする必要はありません。代わりに、既知のテンプレート フォーマットを記述する外部のデータ ファイルを使用することができます。

- 新規機能は、現在の導入環境を損ねることなく、NetFlow に迅速に追加できます。
- バージョン 9 フォーマットは新しいプロトコルや開発中のプロトコルに適応できるため、NetFlow はこれらのプロトコルに対して「将来的に対応」します。

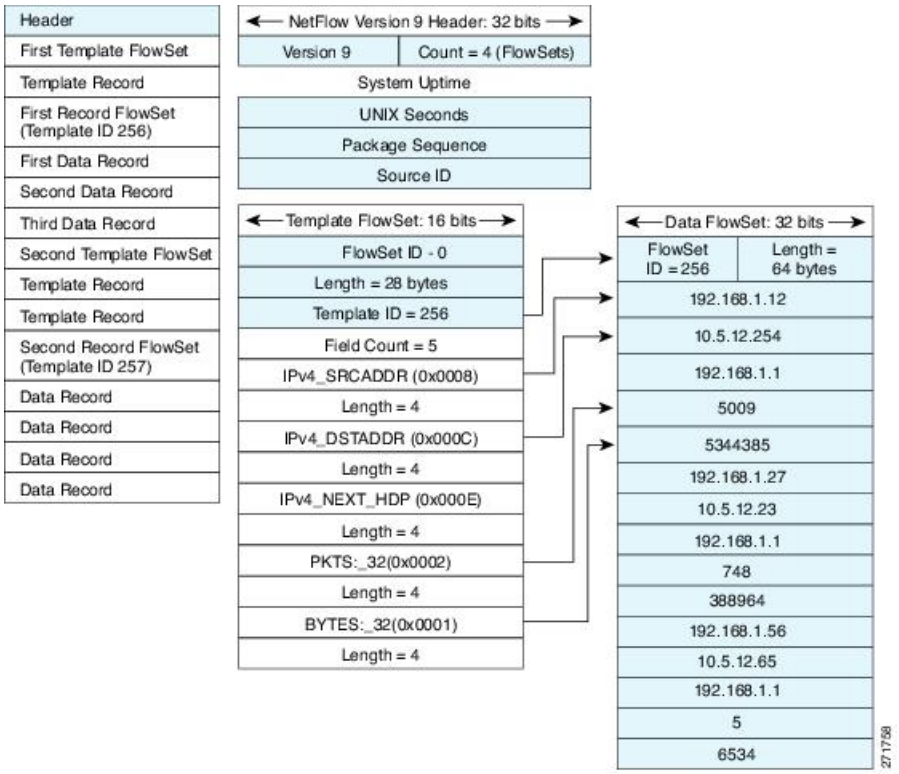
バージョン 9 のエクスポート フォーマットは、パケット ヘッダーとそれに続く 1 つ以上のテンプレート フロー セットまたはデータ フロー セットで構成されています。テンプレート フローセットでは、将来のデータ フローセットに表示されるフィールドの説明が提供されます。このようなデータ フロー セットは、後で同じエクスポート パケットまたは後続のエクスポート パケットで発生する可能性があります。テンプレート フロー セットおよびデータ フロー セットは、次の図に示すように、単一のエクスポート パケットに混在させることができます。

**図 *1* : バージョン *9* エクスポート パケット**




NetFlow Version 9 では、送信されるデータを NetFlow コレクタが理解できるように、テンプレート データを定期的にエクスポートします。また、テンプレートのデータ フロー セットもエクスポートします。Flexible NetFlow の主な利点は、ユーザーがフロー レコードを設定すると、バージョン 9 テンプレートに効率的に変換され、コレクタに転送されることです。下の図に、ヘッダー、テンプレート フロー セットおよびデータ フロー セットを含めて、NetFlow Version 9 エクスポート フォーマットの詳細な例を示します。

図 *2 : NetFlow* バージョン *9* エクスポート フォーマットの詳細例



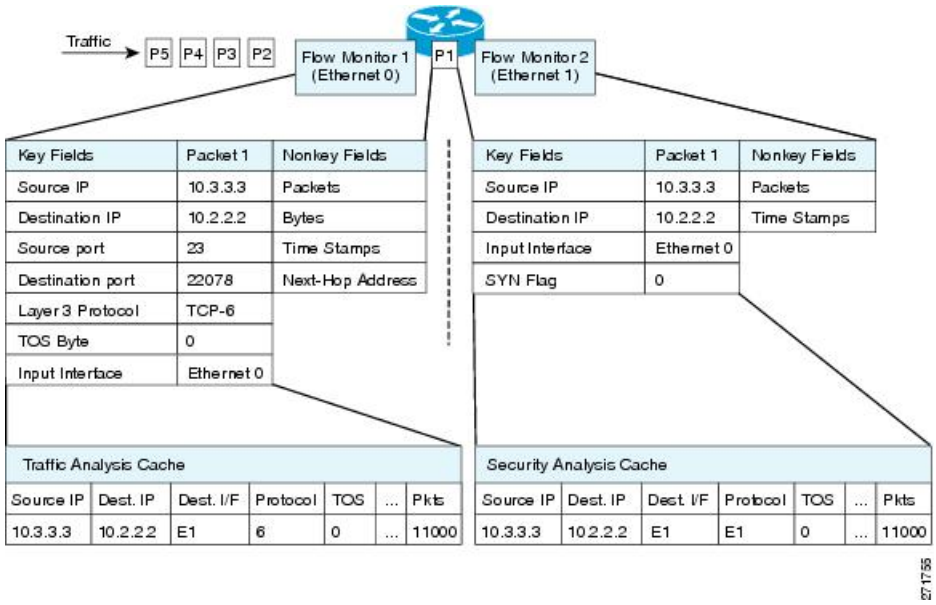

## フロー モニター

フロー モニターは Flexible NetFlow のネットワーク トラフィックの監視を実行するコンポーネントで、インターフェイスに適用されます。

フロー データはネットワーク トラフィックから収集され、フロー レコードの key フィールドおよび nonkey フィールドに基づいて監視プロセス中にフロー モニター キャッシュに追加されます。

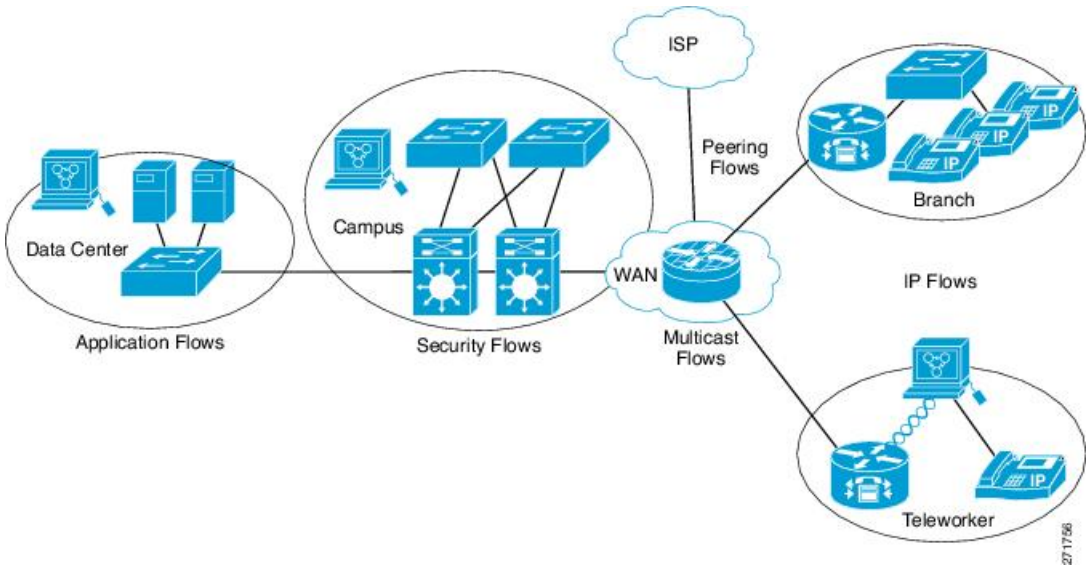
Flexible NetFlow は、同じトラフィックのさまざまなタイプの分析を実行するために使用できます。下の図では、入力インターフェイス上の標準トラフィック分析のために設計されたレコードと、出力インターフェイス上のセキュリティ分析のために設計されたレコードを使用してパケット 1 が分析されます。

図 *3*:*2* つのフロー モニターを使用した同じトラフィックの分析例




下の図に、カスタム レコードを使用して複数のタイプのフロー モニターを適用するより複雑な方法の例を示します。

図 *4*:カスタム レコードでの複数のタイプのフロー モニターの複雑な使用例




## 標準

デフォルトのキャッシュ タイプは「normal」です。このモードでは、キャッシュ内のエントリが timeout active 設定と timeout inactive 設定に従って期限切れになります。キャッシュ エントリは、期限切れになるとキャッシュから削除され、設定されている何らかのエクスポータによってエクスポートされます。

## フロー サンプラー

フローサンプラーは、ルータのコンフィギュレーションで別のコンポーネントとして作成されます。フローサンプラーは、分析用に選択されるパケット数を制限することで、NetFlow Lite を実行しているデバイス上の負荷を削減するために使用されます。

フロー サンプリングでは、ルータのパフォーマンスに対するモニタリング精度が交換されます。サンプラーをフローモニターに適用すると、フローモニターが分析する必要のあるパケット数が減少するため、ルータでフローモニターを実行するためのオーバーヘッド負荷が低下します。フロー モニターで分析されるパケット数が減少すると、フロー モニターのキャッシュに格納される情報の精度が、それに応じて低下します。

**ip flow monitor** コマンドを使用してインターフェイスに適用される場合、サンプラーはフローモニターと組み合わせて使用されます。

# デフォルト設定

次の表に、device の NetFlow Lite デフォルト設定を示します。

**表 *3 : NetFlow Lite* のデフォルト設定**

| 設定 | デフォルト |
|---|---|
| フロー アクティブ タイムアウト | 1800 秒<br>（注） この設定のデフォルト値は特定の NetFlow Lite 設定では高すぎる場合があります。低い値（180 秒または 300 秒）への変更を検討してください。 |
| フロー タイムアウトの非アクティブ化 | イネーブル、30 秒 |
| フロー アップデート タイムアウト | 1800 秒 |
| デフォルト キャッシュ サイズ | 16640 エントリ |

# Flexible NetFlow の設定方法

Flexible Netflow を設定するには、次の一般的な手順に従います。

1. フローにキー フィールドおよび非キー フィールドを指定して、フロー レコードを作成します。
2. プロトコルを指定して任意のフローエクスポータを作成し、宛先ポート、宛先、およびその他のパラメータを転送します。
3. フロー レコードおよびフロー エクスポータに基づいて、フロー モニターを作成します。

4. 任意のサンプラーを作成します。
5. レイヤ 2 ポート、レイヤ 3 ポート、または VLAN にフロー モニターを適用します。

# フロー レコードの作成

フロー レコードを作成し、照合するキー、および収集するフィールドをフロー内に追加できます。

## 手順の概要

1. **configure terminal**
2. **flow record** *name*
3. **description** *string*
4. **match** *type*
5. **collect** *type*
6. **end**
7. **show flow record** [ **name** *record-name*]
8. **copy running-config startup-config**

## 手順の詳細

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 1 | **configure terminal**<br>例：<br>`スイッチ# configure terminal` | グローバル コンフィギュレーション モードを開始します。 |
| ステップ 2 | **flow record** *name*<br>例：<br>`スイッチ(config)# flow record test`<br>`スイッチ(config-flow-record)#` | フロー レコードを作成し、フロー レコード コンフィギュレーション モードを開始します。 |
| ステップ 3 | **description** *string*<br>例：<br>`スイッチ(config-flow-record)# description Ipv4Flow` | （任意）最大 63 文字で、このフローの説明を指定します。 |
| ステップ 4 | **match** *type*<br>例：<br>`スイッチ(config-flow-record)# match ipv4 source` | 一致キーを指定します。 |

| | コマンドまたはアクション | 目的 |
|---|---|---|
| | **address**<br>スイッチ(config-flow-record)# **match ipv4 destination address**<br>スイッチ(config-flow-record)# **match flow direction** | |
| ステップ 5 | **collect** *type*<br>例：<br>スイッチ(config-flow-record)# **collect counter bytes layer2 long**<br>スイッチ(config-flow-record)# **collect counter bytes long**<br>スイッチ(config-flow-record)# **collect timestamp absolute first**<br>スイッチ(config-flow-record)# **collect transport tcp flags**<br>スイッチ(config-flow-record)# **collect interface output** | コレクション フィールドを指定します。<br>（注） フローレコードの collect フィールドとしての **collect interface output** がフローモニターにある場合は、スイッチの宛先アドレスに基づいて出力インターフェイスが検出されます。そのため、他のフロー モニタの場合は、次の設定が必要です。<br>• ipv4 フローモニターの場合は、「**match ip destination address**」を設定します<br>• ipv6 フローモニターの場合は、「**match ipv6 destination address**」を設定します<br>• データリンクフローモニターの場合は、「**match datalink mac output**」を設定します<br>次のアドレスのいずれかにフローが作成された場合、**collect interface output** フィールドに **NULL** の値が返されます。<br>• L3 ブロードキャスト<br>• L2 ブロードキャスト<br>• L3 マルチキャスト<br>• L2 マルチキャスト<br>• L2 の不明な宛先。 |
| ステップ 6 | **end**<br>例：<br>スイッチ(config-flow-record)# **end** | 特権 EXEC モードに戻ります。 |

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 7 | **show flow record** [ **name** *record-name*]<br>例：<br>スイッチ **show flow record test** | （任意）NetFlow のフロー レコード情報を表示します。 |
| ステップ 8 | **copy running-config startup-config**<br>例：<br>スイッチ# **copy running-config startup-config** | （任意）コンフィギュレーション ファイルに設定を保存します。 |

### 次のタスク

エクスポート フォーマット、プロトコル、宛先、およびその他のパラメータを指定することによって、任意でフロー エクスポータを定義します。

# フロー エクスポータの作成

フロー エクスポートを作成して、フローのエクスポート パラメータを定義できます。

（注） フロー エクスポータごとに、1 つの宛先のみがサポートされます。複数の宛先にデータをエクスポートする場合は、複数のフロー エクスポータを設定してフロー モニターに割り当てる必要があります。

IPv4 アドレスを使用して宛先にエクスポートできます。

### 手順の概要

1. **configure terminal**
2. **flow exporter** *name*
3. **description** *string*
4. **destination** {*ipv4-address*}
5. **dscp** *value*
6. **transport udp** *number*
7. **ttl** *seconds*
8. **export-protocol** {**netflow-v9**}
9. **end**
10. **show flow exporter** [ **name** *record-name*]
11. **copy running-config startup-config**

手順の詳細

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 1 | **configure terminal**<br>例：<br>`スイッチ# configure terminal` | グローバル コンフィギュレーション モードを開始します。 |
| ステップ 2 | **flow exporter** *name*<br>例：<br>`スイッチ(config)# flow exporter ExportTest` | フロー エクスポータを作成し、フロー エクスポータ コンフィギュレーション モードを開始します。 |
| ステップ 3 | **description** *string*<br>例：<br>`スイッチ(config-flow-exporter)# description ExportV9` | （任意）最大 63 文字で、このフローの説明を指定します。 |
| ステップ 4 | **destination** {*ipv4-address*}<br>例：<br>`スイッチ(config-flow-exporter)# destination 192.0.2.1 (IPv4 destination)` | このエクスポータに IPv4 宛先アドレスまたはホスト名を設定します。 |
| ステップ 5 | **dscp** *value*<br>例：<br>`スイッチ(config-flow-exporter)# dscp 0` | （任意）DSCP（DiffServ コードポイント）値を指定します。範囲は 0 ～ 63 です。デフォルトは 0 です。 |
| ステップ 6 | **transport udp** *number*<br>例：<br>`スイッチ(config-flow-exporter)# transport udp 200` | （任意）NetFlow コレクタに到達するために使用する UDP ポートを指定します。 |
| ステップ 7 | **ttl** *seconds*<br>例：<br>`スイッチ(config-flow-exporter)# ttl 210` | （任意）エクスポータによって送信されるデータグラムの存続可能時間（TTL）値を設定します。範囲は 1 ～ 255 秒です。デフォルトは 255 です。 |

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 8 | **export-protocol** {**netflow-v9**}<br>例：<br>`スイッチ(config-flow-exporter)# export-protocol netflow-v9` | エクスポータで使用される NetFlow エクスポート プロトコルのバージョンを指定します。 |
| ステップ 9 | **end**<br>例：<br>`スイッチ(config-flow-record)# end` | 特権 EXEC モードに戻ります。 |
| ステップ 10 | **show flow exporter** [ **name** *record-name*]<br>例：<br>`スイッチ# show flow exporter ExportTest` | （任意）NetFlow のフロー エクスポータ情報を表示します。 |
| ステップ 11 | **copy running-config startup-config**<br>例：<br>`スイッチ# copy running-config startup-config` | （任意）コンフィギュレーション ファイルに設定を保存します。 |

### 次のタスク

フロー レコードおよびフロー エクスポータに基づいて、フロー モニターを定義します。

# フロー モニターの作成

フロー モニタを作成して、フロー レコードおよびフロー エクスポータと関連付けることができます。

### 手順の概要

1. **configure terminal**
2. **flow monitor** *name*
3. **description** *string*
4. **exporter** *name*
5. **record** *name*
6. **cache** { **timeout** {**active** | **inactive**} *seconds* | **type normal** }
7. **end**
8. **show flow monitor** [ **name** *record-name*]
9. **copy running-config startup-config**

手順の詳細

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 1 | **configure terminal**<br>例：<br>`スイッチ# configure terminal` | グローバル コンフィギュレーション モードを開始します。 |
| ステップ 2 | **flow monitor** *name*<br>例：<br>`スイッチ(config)# flow monitor MonitorTest`<br>`スイッチ (config-flow-monitor)#` | フローモニタを作成し、フローモニタ コンフィギュレーション モードを開始します。 |
| ステップ 3 | **description** *string*<br>例：<br>`スイッチ(config-flow-monitor)# description Ipv4Monitor` | （任意）最大 63 文字で、このフローの説明を指定します。 |
| ステップ 4 | **exporter** *name*<br>例：<br>`スイッチ(config-flow-monitor)# exporter ExportTest` | フロー エクスポータとこのフロー モニタを関連付けます。 |
| ステップ 5 | **record** *name*<br>例：<br>`スイッチ(config-flow-monitor)# record test` | フロー レコードを指定したフロー モニタと関連付けます。 |
| ステップ 6 | **cache** { **timeout** {**active** \| **inactive**} *seconds* \| **type normal** }<br>例：<br>`スイッチ(config-flow-monitor)# cache timeout active 15000` | 指定したフロー モニタとフロー キャッシュを関連付けます。 |
| ステップ 7 | **end**<br>例：<br>`スイッチ(config-flow-monitor)# end` | 特権 EXEC モードに戻ります。 |

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 8 | **show flow monitor** [ **name** *record-name*]<br>例：<br>スイッチ **show flow monitor name MonitorTest** | （任意）NetFlow のフロー モニタ情報を表示します。 |
| ステップ 9 | **copy running-config startup-config**<br>例：<br>スイッチ# **copy running-config startup-config** | （任意）コンフィギュレーション ファイルに設定を保存します。 |

### 次のタスク

レイヤ 2 インターフェイス、レイヤ 3 インターフェイス、または VLAN にフロー モニタを適用します。

# サンプラーの作成

サンプラーを作成し、フローの NetFlow サンプリング レートを定義できます。

### 手順の概要

1. **configure terminal**
2. **sampler** *name*
3. **description** *string*
4. **mode** {**deterministic** {*m* - *n*} | **random** {*m* - *n*}}
5. **end**
6. **show sampler** [*name*]
7. **copy running-config startup-config**

### 手順の詳細

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 1 | **configure terminal**<br>例：<br>スイッチ# **configure terminal** | グローバル コンフィギュレーション モードを開始します。 |
| ステップ 2 | **sampler** *name*<br>例： | サンプラーを作成し、サンプラー コンフィギュレーション モードを開始します。 |

| | コマンドまたはアクション | 目的 |
|---|---|---|
| | スイッチ(config)# **sampler SampleTest** | |
| ステップ 3 | **description** *string*<br>例：<br>スイッチ(config-flow-sampler)# **description samples** | （任意）最大 63 文字で、このフローの説明を指定します。 |
| ステップ 4 | **mode** {**deterministic** {*m* - *n*} \| **random** {*m* - *n*}}<br>例：<br>スイッチ(config-flow-sampler)# **mode random 1 out-of 1022** | ランダム サンプル モードを定義します。<br>インターフェイスに対してランダムサンプラーまたは確定的サンプラーのいずれも設定できます。*n* パケット ウィンドウから *m* 個のパケットを選択します。ウィンドウ サイズには、32～1022 の範囲のパケットを選択します。<br>インターフェイスにサンプラーを設定する際は、次の点に注意してください。<br>• 確定的サンプラー（s1 など）を使用してモニターを接続する場合、同じサンプラーs1との接続ごとにdevice（ハードウェア）から 4 つの使用可能なサンプラーのうちの新しい空きサンプラーの 1 つを使用します。したがって、サンプラーとモニターの接続は、4 つを超えて行うことができません。<br>• これとは逆に、ランダムサンプラー（たとえば、この場合もs1など）を使用してモニターを接続する場合、最初の接続だけが device（ハードウェア）の新しいサンプラーを使用します。同じサンプラーs1を使用するすべての接続のうちの残りは同じサンプラーを共有します。<br>• この動作により、確定的サンプラーを使用する際は、サンプリング レートとdeviceが何を送信するかを比較して、適切な数のフローがサンプリングされているかを常に確認することができます。複数のインターフェイスに同じランダムサンプラーを使用している場合は、インターフェイスからのフローを常にサンプリングすることができ、他のインターフェイスからのフローは常にスキップできます。 |

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 5 | **end**<br>例：<br>スイッチ(config-flow-sampler)# **end** | 特権 EXEC モードに戻ります。 |
| ステップ 6 | **show sampler** [*name*]<br>例：<br>スイッチ **show sample SampleTest** | （任意）NetFlow サンプラに関する情報を表示します。 |
| ステップ 7 | **copy running-config startup-config**<br>例：<br>スイッチ# **copy running-config startup-config** | （任意）コンフィギュレーション ファイルに設定を保存します。 |

### 次のタスク

ソースインターフェイス、または SVI にフローモニターを適用します。

# インターフェイスへのフローの適用

フロー モニターおよびオプションのサンプラーをインターフェイスに適用できます。

### 手順の概要

1. **configure terminal**
2. **interface** *type*
3. {**ip flow monitor** | **ipv6 flow monitor**}*name* [| **sampler** *name*] {**input**}
4. **end**
5. **show flow interface** [*interface-type number*]
6. **copy running-config startup-config**

### 手順の詳細

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 1 | **configure terminal**<br>例：<br>スイッチ# **configure terminal** | グローバル コンフィギュレーション モードを開始します。 |

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 2 | **interface** *type*<br>例：<br>`スイッチ(config)# interface GigabitEthernet1/0/1` | インターフェイス コンフィギュレーション モードを開始し、インターフェイスを設定します。 |
| ステップ 3 | {**ip flow monitor** \| **ipv6 flow monitor**}*name* [\| **sampler** *name*] {**input**}<br>例：<br>`スイッチ(config-if)# ip flow monitor MonitorTest input` | 入力または出力パケットに対応するインターフェイスに、IPv4 または IPv6 フロー モニター、およびオプションのサンプラーを関連付けます。<br>入力と出力の両方向でインターフェイスに複数のモニターを関連付けることができます。 |
| ステップ 4 | **end**<br>例：<br>`スイッチ(config-flow-monitor)# end` | 特権 EXEC モードに戻ります。 |
| ステップ 5 | **show flow interface** [*interface-type number*]<br>例：<br>`スイッチ# show flow interface` | （任意）インターフェイスの NetFlow 情報を表示します。 |
| ステップ 6 | **copy running-config startup-config**<br>例：<br>`スイッチ# copy running-config startup-config` | （任意）コンフィギュレーション ファイルに設定を保存します。 |

# VLAN 上でのブリッジ型 NetFlow の設定

フロー モニターおよびオプションのサンプラーを VLAN に適用できます。

### 手順の概要

1. **configure terminal**
2. **vlan** [**configuration**] *vlan-id*
3. **ip flow monitor** *monitor name* [ **sampler** *sampler name*] { **input** }
4. **copy running-config startup-config**

手順の詳細

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 1 | **configure terminal**<br>例：<br>`スイッチ# configure terminal` | グローバル コンフィギュレーション モードを開始します。 |
| ステップ 2 | **vlan** [**configuration**] *vlan-id*<br>例：<br>`スイッチ(config)# vlan configuration 30`<br>`スイッチ(config-vlan-config)#` | VLAN または VLAN コンフィギュレーション モードを開始します。 |
| ステップ 3 | **ip flow monitor** *monitor name* [ **sampler** *sampler name*] { **input** }<br>例：<br>`スイッチ(config-vlan-config)# ip flow monitor MonitorTest input` | 入力パケットに対応する VLAN に、フローモニターおよびオプションのサンプラーを関連付けます。 |
| ステップ 4 | **copy running-config startup-config**<br>例：<br>`スイッチ# copy running-config startup-config` | （任意）コンフィギュレーションファイルに設定を保存します。 |

# レイヤ 2 NetFlow の設定

NetFlow Lite レコード内でレイヤ 2 キーを定義できます。このレコードを使用して、レイヤ 2 インターフェイスのフローをキャプチャできます。

手順の概要

1. **configure terminal**
2. **flow record** *name*
3. **match datalink**{**ethertype** | **mac**{**destination** {**address input**} | **source**{**address input**}}}
4. **end**
5. **show flow record** [*name*]
6. **copy running-config startup-config**

手順の詳細

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 1 | **configure terminal**<br>例：<br>`スイッチ# configure terminal` | グローバル コンフィギュレーション モードを開始します。 |
| ステップ 2 | **flow record** *name*<br>例：<br>`スイッチ(config)# flow record L2_record`<br>`スイッチ(config-flow-record)#` | フロー レコード コンフィギュレーション モードを開始します。 |
| ステップ 3 | **match datalink**{**ethertype** \| **mac**{**destination** {**address input**} \| **source**{**address input**}}}<br>例：<br>`スイッチ(config-flow-record)# match datalink mac source address input`<br>`スイッチ(config-flow-record)# match datalink mac destination address input` | レイヤ 2 属性をキーとして指定します。この例では、入力時のパケットの送信元および宛先の MAC アドレスがキーです。<br>（注） データリンク フロー モニターがインターフェイスまたは VLAN レコードに割り当てられている場合、非 IPv4 または非 IPv6 トラフィック用のフローだけが作成されます。 |
| ステップ 4 | **end**<br>例：<br>`スイッチ(config-flow-record)# end` | 特権 EXEC モードに戻ります。 |
| ステップ 5 | **show flow record** [*name*]<br>例：<br>`スイッチ# show flow record` | （任意）インターフェイスの NetFlow 情報を表示します。 |
| ステップ 6 | **copy running-config startup-config**<br>例：<br>`スイッチ# copy running-config startup-config` | （任意）コンフィギュレーション ファイルに設定を保存します。 |

# Flexible NetFlow の監視

次の表にあるコマンドを使用して、Flexible NetFlow をモニタリングできます。

**表 *4 : Flexible NetFlow* のモニタリング コマンド**

| コマンド | 目的 |
|---|---|
| **show flow exporter** [**broker** \| **export-ids** \| **name** \| *name* \| **statistics** \| **templates**] | NetFlow のフロー エクスポータ情報と統計情報を表示します。 |
| **show flow exporter** [ **name** *exporter-name*] | NetFlow のフロー エクスポータ情報と統計情報を表示します。 |
| **show flow interface** | NetFlowインターフェイスに関する情報を表示します。 |
| | NetFlow のフロー モニター情報と統計情報を表示します。 |
| **show flow monitor statistics** | フロー モニターの統計情報を表示します。 |
| | 指定された形式でフローモニターのキャッシュの内容を表示します。 |
| **show flow record** [ **name** *record-name*] | NetFlowのフロー レコード情報を表示します。 |
| **show sampler** [**broker** \| **name** \| *name*] | NetFlow サンプラーに関する情報を表示します。 |

# 設定例 NetFlow Lite

## 例：フローの設定

（注） フローを設定する場合、フロー レコードで定義されたプロトコル、送信元ポート、宛先ポート、最初と最後のタイムスタンプ、パケットおよびバイト カウンタが必要です。これらがないと、「Warning: Cannot set protocol distribution with this Flow Record. Require protocol, source and destination ports, first and last timestamps and packet and bytes counters.」というエラー メッセージが表示されます。

フローを作成し、そのフローをインターフェイスに適用する例を示します。

```
スイッチ# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.

スイッチ(config)# flow exporter export1
スイッチ(config-flow-exporter)# destination 10.0.101.254
スイッチ(config-flow-exporter)# transport udp 2055
スイッチ(config-flow-exporter)# template data timeout 60
スイッチ(config-flow-exporter)# exit
スイッチ(config)# flow record record1
スイッチ(config-flow-record)# match ipv4 source address
スイッチ(config-flow-record)# match ipv4 destination address
スイッチ(config-flow-record)# match ipv4 protocol
スイッチ(config-flow-record)# match transport source-port
スイッチ(config-flow-record)# match transport destination-port
スイッチ(config-flow-record)# collect counter bytes long
スイッチ(config-flow-record)# collect counter packets long
スイッチ(config-flow-record)# collect timestamp sys-uptime first
スイッチ(config-flow-record)# collect timestamp sys-uptime last
スイッチ(config-flow-record)# exit
スイッチ(config)# sampler SampleTest
スイッチ(config-sampler)# mode random 1 out-of 100
スイッチ(config-sampler)# exit
スイッチ(config)# flow monitor monitor1
スイッチ(config-flow-monitor)# cache timeout active 300
スイッチ(config-flow-monitor)# cache timeout inactive 120
スイッチ(config-flow-monitor)# record record1
スイッチ(config-flow-monitor)# exporter export1
スイッチ(config-flow-monitor)# exit
スイッチ(config)# interface GigabitEthernet1/0/1
スイッチ(config-if)# ip flow monitor monitor1 sampler SampleTest input
スイッチ(config-if)# end
```

# Flexible NetFlow の機能情報

| リリース | 変更内容 |
|---|---|
| Cisco IOS Release 15.2(3)E | この機能が導入されました。 |
| Cisco IOS XE Gibraltar 16.12.1 | IPv6 トラフィックについて、FNF の SGT フィールドと DGT フィールドのサポートが導入されました。 |

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。