



## パフォーマンス管理の実装

Cisco IOS XR ソフトウェアのパフォーマンス管理（PM）では、次のタスクを実行するためのフレームワークが提供されます。

- データを保管および取得するために PM 統計情報を収集して TFTP サーバにエクスポートする
- 拡張マークアップ言語（XML）のクエリを使用してシステムをモニタする
- しきい値条件が一致するときにシステム ログメッセージを生成するしきい値条件を設定する

PMシステムでは、システムリソースの使用率をグラフ化して、容量を計画したり、トラフィックエンジニアリングに使用したり、傾向を分析したりするために役立つデータを収集します。

- [パフォーマンス管理を実装する前提条件](#)（1 ページ）
- [パフォーマンス管理の実装に関する情報](#)（2 ページ）
- [PM 機能の概要](#)（2 ページ）
- [PM の利点](#)（3 ページ）
- [PM 統計情報収集の概要](#)（3 ページ）
- [パフォーマンス管理の実装方法](#)（8 ページ）

## パフォーマンス管理を実装する前提条件

ネットワーク オペレーションセンター（NOC）にパフォーマンス管理を導入する前に、次の前提条件を満たしていることを確認します。

- 管理ソフトウェアのパッケージインストールエンベロップ（PIE）インストールしてアクティブにする必要があります。
- 適切なタスク ID を含むタスク グループに関連付けられているユーザ グループに属している必要があります。このコマンドリファレンスには、各コマンドに必要なタスク ID が含まれます。ユーザ グループの割り当てが原因でコマンドを使用できないと考えられる場合、AAA 管理者に連絡してください。
- TFTP サーバへの接続が必要です。

# パフォーマンス管理の実装に関する情報

## PM 機能の概要

パフォーマンス管理（PM）フレームワークは次の2つの主要なコンポーネントで構成されています。

- PM 統計情報サーバ
- PM 統計情報収集機能

## PM 統計情報サーバ

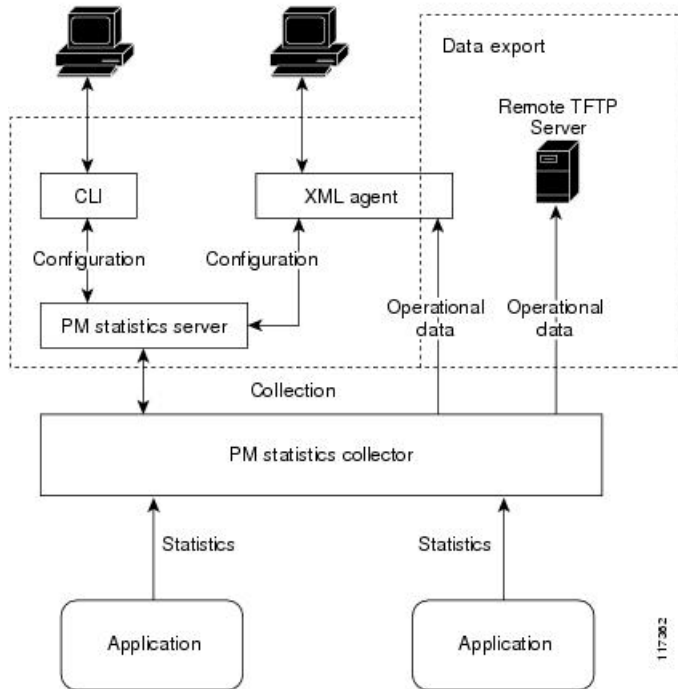
PM 統計情報サーバは統計情報収集、エンティティインスタンスモニタリング収集、しきい値モニタリングのフロントエンドです。コマンドラインインターフェイス（CLI）または XML スキームから設定されたすべての PM 統計情報収集およびしきい値条件は、PM 統計情報サーバによって処理され、PM 統計情報機能に分散されます。

## PM 統計情報収集機能

PM 統計情報収集機能ではエンティティインスタンスから統計情報を収集し、そのデータをメモリに格納します。プロセスの再起動時に情報を利用できるように、メモリの内容のチェックポイントが行われます。さらに、PM 統計情報の収集機能は、XML エージェントおよび TFTP サーバへの動作データのエクスポートを担当します。

[図 1: PM コンポーネントの通信 \(3 ページ\)](#) は、PM システムを構成するコンポーネントの関係を表しています。

図 1: PM コンポーネントの通信



## PM の利点

PM システムには次の利点があります。

- データ収集ポリシーを設定可能
- TFTP を経由してバイナリ形式で統計データを効率的に転送
- エンティティ インスタンス モニタリングをサポート
- しきい値モニタリングをサポート
- プロセスの再起動時およびプロセッサのフェールオーバー時にデータの一貫性を確保

## PM 統計情報収集の概要

PM 統計情報収集では、はじめに PM システム内にあるエンティティのすべてのインスタンスに関連付けられているすべての属性から統計情報を収集します。次に、統計データをバイナリファイル形式で TFTP サーバにエクスポートします。たとえば、マルチプロトコル ラベル スイッチング (MPLS) ラベル配布プロトコル (LDP) 統計情報収集では、ルータ上のすべての MPLS LDP セッションに関連付けられているすべての属性から統計データを収集します。

この表では、PM システムのエンティティおよび関連インスタンスを一覧で示します。

表 1: エンティティ クラスおよび関連付けられているインスタンス

エンティティ クラス	インスタンス
BGP	ネイバーまたはピア
インターフェイス基本カウンタ	インターフェイス
インターフェイスデータレート	インターフェイス
インターフェイス汎用カウンタ	インターフェイス
MPLS LDP	LDP セッション
ノード CPU	ノード
ノード メモリ	ノード
ノード プロセス	プロセス
OSPFv2	プロセス
OSPFv3	プロセス



(注) PMシステムを構成するエンティティに関連付けられているすべての属性のリストについては、「表 4: 属性と値 (12 ページ)」を参照してください。



(注) インターフェイスタイプに応じて、インターフェイスはインターフェイスの汎用カウンタまたは基本カウンタのいずれかをサポートします。インターフェイスの基本カウンタをサポートするインターフェイスは、インターフェイスのデータ レートをサポートしません。

## PM 統計情報をエクスポートするためのバイナリ ファイル形式

次のサンプルでは、バイナリ ファイル形式を説明します。

```
Version : 4 Bytes
NoOf Entities : 1 Byte (e.g. . 4 )
Entity Identifier : 1 Byte (e.g NODE=1,Interface=2,BGP=3)
Options : 2 Bytes
NoOf SubEntities : 1 Byte (2)
SubEntity Identifier : 1 Byte (e.g BGP-PEERS )
Time Stamp 4 Bytes (Reference Time : Start Ref Time)
No Of Instances : 2 Byte (e.g 100)
Key Instance : Variable
```

```

NoOfSamples: 1 Byte (e.g 10 Samples)
SampleNo : 1 Byte (e.g Sample No 1)
Time Stamp 4 Bytes (Sample Time)
StatCounterName :1 Byte (PeerSessionsEst=1)
StatCounterValue :8 Bytes ( for all counters)
Repeat for Each StatCounterName
Repeat for Each Sample No(Time Interval)
Repeat for All Instances
Repeat for All SubTypes
Repeat for All Entities

```

## エンティティのバイナリ ファイル ID 割り当て、サブエンティティ、統計情報カウンタ名

この表では、バイナリ ファイルでのさまざまな値の割り当ておよびキーを説明します。

表 2: バイナリ形式の値とキー

エンティティ	サブエンティティ	キー	統計情報カウンタ
ノード (1)	CPU (1)	CPU キー <Node ID>	「表 3 : エンティティとサブエンティティでサポートされる統計情報カウンタ (6 ページ)」を参照してください
	メモリ (2)	メモリ キー <Node ID>	
	プロセス (3)	ノード プロセス キー <NodeProcessID>	
インターフェイス (2)	汎用カウンタ (1)	汎用カウンタ キー <ifName>	
	データ レート カウンタ (2)	データ レート カウンタ キー <ifName>	
	基本カウンタ (3)	基本カウンタ キー <ifName>	
BGP (3)	ピア (1)	ピア キー <IpAddress>	
MPLS (4)	予約済み (1)	—	
	予約済み (2)	—	
	LDP (4)	LDP セッション キー <IpAddress>	
OSPF (5)	v2protocol (1)	イ ン ス タ ン ス <process_instance>	

エンティティ	サブエンティティ	キー	統計情報カウンタ
	v3protocol (2)	インスタンス <process_instance	



- (注) <ifName> : 長さの値は変数です。最初の2バイトにはインスタンスIDのサイズが含まれます。その次にインスタンス ID 文字列 (インターフェイス名) が続きます。
- <IpAddress> : IP アドレスが含まれる 4 バイトです。
- <NodeProcessID> : 64 ビットのインスタンス ID です。最初の 32 ビットにはノード ID が含まれ、次の 32 ビットにはプロセス ID が含まれます。
- <NodeID> : ノード ID が含まれる 32 ビット インスタンスです。
- <process\_instance> : 長さの値は変数です。最初の 2 バイトにはインスタンス ID のサイズが含まれ、その次にインスタンス ID 文字列 (プロセス名) が続きます。



- (注) 括弧の中の数字 (「表 2: バイナリ形式の値とキー (5 ページ)」の各エンティティとサブエンティティに関連付けられている数字) は、TFTP ファイルに表示されるエンティティ ID とサブエンティティ ID を表します。

この表では、エンティティとサブエンティティのバイナリ ファイルに収集される、サポート対象の統計情報カウンタを説明します。

表 3: エンティティとサブエンティティでサポートされる統計情報カウンタ

エンティティ	サブエンティティ	統計情報カウンタ
ノード (1)	CPU (1)	AverageCPUUsed、NoProcesses
	メモリ (2)	CurrMemory、PeakMemory
	プロセス (3)	PeakMemory、AverageCPUUsed、NoThreads
インターフェイス (2)	汎用カウンタ (1)	InPackets、InOctets、OutPackets、OutOctets、InUcastPkts、InMulticastPkts、InBroadcastPkts、OutUcastPkts、OutMulticastPkts、OutBroadcastPkts、OutputTotalDrops、InputTotalDrops、InputQueueDrops、InputUnknownProto、OutputTotalErrors、OutputUnderrun、InputTotalErrors、InputCRC、InputOverrun、InputFrame
	データ レート カウンタ (2)	InputDataRate、InputPacketRate、OutputDataRate、OutputPacketRate、InputPeakRate、InputPeakPkts、OutputPeakRate、OutputPeakPkts、Bandwidth

エンティティ	サブエンティティ	統計情報カウンタ
	基本カウンタ	InPackets、InOctets、OutPackets、OutOctets、InputTotalDrops、InputQueueDrops、InputTotalErrors、OutputTotalErrors、OutputQueueDrops、OutputTotalErrors
BGP (3)	ピア (1)	InputMessages、OutputMessages、InputUpdateMessages、OutputUpdateMessages、ConnEstablished、ConnDropped、ErrorsReceived、ErrorsSent
MPLS (4)	LDP (4)	TotalMsgsSent、TotalMsgsRcvd、InitMsgsSent、InitMsgsRcvd、AddressMsgsSent、AddressMsgsRcvd、AddressWithdrawMsgsSent、AddressWithdrawMsgsRcvd、LabelMappingMsgsSent、LabelMappingMsgsRcvd、LabelWithdrawMsgsSent、LabelWithdrawMsgsRcvd、LabelReleaseMsgsSent、LabelReleaseMsgsRcvd、NotificationMsgsSent、NotificationMsgsRcvd、KeepAliveMsgsSent、KeepAliveMsgsRcvd
OSPF (5)	v2protocol (1)	InputPackets、OutputPackets、InputHelloPackets、OutputHelloPackets、InputDBDs、InputDBDsLSA、OutputDBDs、OutputDBDsLSA、InputLSRequests、InputLSRequestsLSA、OutputLSRequests、OutputLSRequestsLSA、InputLSAUpdates、InputLSAUpdatesLSA、OutputLSAUpdates、OutputLSAUpdatesLSA、InputLSAAcks、InputLSAAcksLSA、OutputLSAAcks、OutputLSAAcksLSA、ChecksumErrors
	v3protocol (2)	InputPackets、OutputPackets、InputHelloPackets、OutputHelloPackets、InputDBDs、InputDBDsLSA、OutputDBDs、OutputDBDsLSA、InputLSRequests、InputLSRequestsLSA、OutputLSRequests、OutputLSRequestsLSA、InputLSAUpdates、InputLSAUpdatesLSA、OutputLSAUpdates、OutputLSAUpdatesLSA、InputLSAAcks、InputLSAAcksLSA、OutputLSAAcks、OutputLSAAcksLSA

## バイナリファイルに適用されるファイルの命名規則

次のファイルの命名規則は、TFTP サーバに設定されているディレクトリの場所に送信される PM 統計情報収集に適用されます。

<LR\_NAME>\_<EntityName>\_<SubentityName>\_<TimeStamp>

# パフォーマンス管理の実装方法

## PM 統計情報収集用の外部 TFTP サーバまたはローカルディスクの設定

PM 統計データを外部の TFTP サーバにエクスポートしたり、ローカルファイルシステムにダンプしたりすることができます。ローカル宛先と TFTP 宛先はともに相互に排他的で、一度に設定できるのはどちらか一方です。

### 設定例

次の例では、PM 統計情報収集用に外部の TFTP サーバを設定します。

```
RP/0/RP0/CPU0:Router# configure
RP/0/RP0/CPU0:Router(config)# performance-mgmt resources tftp-server 10.3.40.161 directory
  mypdata/datafiles
RP/0/RP0/CPU0:Router(config)# commit
```

次の例では、PM 統計情報収集用にローカルディスクを設定します。

```
RP/0/RP0/CPU0:Router# configure
RP/0/RP0/CPU0:Router(config)# performance-mgmt resources dump local
RP/0/RP0/CPU0:Router(config)# commit
```

## PM 統計情報収集テンプレートの設定

PM 統計情報収集は、PM 統計情報収集テンプレートから設定されます。PM 統計情報収集テンプレートには、エンティティ、サンプル間隔、TFTP サーバにデータをエクスポートするまでに実行されるサンプリング動作の回数が含まれます。PM 統計情報収集テンプレートがイネーブルの場合、PM 統計情報収集は、テンプレートに設定されているエンティティに関連付けられているすべてのインスタンスからの属性の統計情報を収集します。特定のエンティティに複数のテンプレートを定義できます。ただし、特定のエンティティに一度にイネーブルにできる PM 統計情報テンプレートは 1 つだけです。

### PM 統計情報収集テンプレートを設定するガイドライン

PM 統計情報テンプレートの作成時は、次のガイドラインに従ってください。

- リモート TFTP サーバやローカルディスクに統計情報データをエクスポートする場合、TFTP サーバリソースやローカルダンプリソースを設定する必要があります。
- 特定のエンティティに対して複数のテンプレートを定義できますが、特定のエンティティに対して同時に有効にできる PM 統計情報収集テンプレートは 1 つだけです。
- テンプレートを設定するときは、デフォルトのキーワードを使用してエンティティのテンプレートをデフォルトのテンプレートとして指定するか、またはテンプレートに名前をつけることができます。デフォルトテンプレートには、次のデフォルト値が含まれています。
  - 10 分のサンプル間隔。



- 5つのサンプリング動作のサンプル サイズ。
- サンプル間隔は、サンプリングサイクル中に実行されるサンプリング動作の頻度を設定します。 **sample-interval** コマンドを使用して、サンプルの間隔を設定できます。範囲は1～60分です。
- **sample size** では、データを TFTP サーバにエクスポートする前に実行されるサンプリング動作の数を設定します。 **sample-size** コマンドを使用して、サンプルのサイズを設定できます。範囲は1～60サンプルです。



(注) 小さいサンプル間隔を指定すると CPU 使用率が増加し、大きいサンプル サイズを指定するとメモリ使用率が増加します。そのため、システムのオーバーロードを防ぐために、サンプルサイズとサンプル間隔の調整が必要になる場合があります。

- エクスポート サイクルでは、PM 統計情報収集データが TFTP サーバにエクスポートされる頻度を決定します。エクスポート サイクルは、サンプル間隔にサンプル サイズを掛け合わせて計算します (サンプル間隔 x サンプル サイズ = エクスポート サイクル)。
- テンプレートをイネーブルにすると、**performance-mgmt apply statistics** コマンドの **no** 形式でテンプレートをディセーブルにするまで、サンプリングとエクスポートのサイクルは継続されます。
- 次のエンティティの PM 統計情報収集を有効または無効にするときは、**location** コマンドを使用してノードを指定するか、または **location all** コマンドを使用してすべてのノードの PM 統計情報収集を有効にする必要があります。
  - ノード CPU
  - ノード メモリ
  - ノード プロセス

### 設定例

次の例では、PM 統計情報収集テンプレートの作成方法およびイネーブル方法を示します。

```
RP/0/RP0/CPU0:Router# configure
RP/0/RP0/CPU0:Router(config)# performance-mgmt statistics interface generic-counters
template template 1
RP/0/RP0/CPU0:Router(config)# performance-mgmt statistics interface generic-counters
template 1 sample-size 10
RP/0/RP0/CPU0:Router(config)# performance-mgmt statistics interface generic-counters
template 1 sample-interval 5
RP/0/RP0/CPU0:Router(config)# performance-mgmt apply statistics interface generic-counters
1
RP/0/RP0/CPU0:Router# commit
```

## PM エンティティ インスタンス モニタリングのイネーブル化

エンティティ インスタンス モニタリングでは、特定のエンティティ インスタンスに関連付けられている属性から統計情報を収集します。エンティティ インスタンスのモニタリングがイ

ネーブルな場合、PMシステムは指定したエンティティインスタンスに関連する属性の統計情報だけを収集します。PMシステムでは、モニタリング対象のエンティティのPM統計情報収集テンプレートで設定されているサンプリングサイクルを使用します。ただし、エンティティインスタンスモニタリングは、PM統計情報収集のプロセスとは別のプロセスです。そのため、PM統計情報収集とは連携しません。さらに、エンティティインスタンスモニタリング収集からのデータはPM統計情報収集から独立しています。PM統計情報収集とは異なり、エンティティインスタンスモニタリングからのデータはTFTPサーバにエクスポートされません。各エンティティインスタンスに関連付けられているすべての属性とコマンドの詳細については、「[パフォーマンス管理：詳細（11ページ）](#)」を参照してください。

### 設定例

次の例に、ノードCPUエンティティインスタンスのエンティティインスタンスモニタリングを有効にする方法を示します。

```
RP/0/RP0/CPU0:Router# configure
RP/0/RP0/CPU0:Router(config)# performance-mgmt apply monitor node cpu location 0/RP0/CPU0
default
RP/0/RP0/CPU0:Router(config)# commit
```

## PM しきい値モニタリングテンプレートの設定

PMシステムでは、しきい値条件を設定して、しきい値違反の属性をモニタできます。しきい値条件はPMしきい値モニタリングテンプレートから設定されます。PMしきい値テンプレートがイネーブルの場合、PMシステムはテンプレートに設定されているしきい値条件の属性のインスタンスをすべてモニタします。サンプル間隔の最後にしきい値条件が一致すると、PMシステムではしきい値条件に一致したそれぞれのインスタンスにシステムログメッセージを生成します。すべてのエンティティの属性および各属性に関連付けられている値の範囲のリストについては、[パフォーマンス管理：詳細（11ページ）](#)を参照してください。

### PM しきい値モニタリングテンプレートを設定するガイドライン

PMしきい値モニタリングテンプレートを設定するときには、次のガイドラインに従います。

- テンプレートが有効になると、**performance-mgmt apply thresholds** コマンドの **no** 形式でテンプレートが無効になるまで、しきい値モニタリングが継続されます。
- 1つのエンティティで一度にイネーブルにできるPMしきい値テンプレートは1つだけです。
- 次のエンティティに対しPMしきい値モニタリングテンプレートを有効または無効にするときは、**location** コマンドを使用してノードを指定するか、または **location all** コマンドを使用してすべてのノードのPM統計情報収集を有効にする必要があります。
  - ノード CPU
  - ノード メモリ
  - ノード プロセス

### 設定例

この例では、PM しきい値モニタリングテンプレートの作成方法およびイネーブル方法を示します。この例では、ノード CPU エンティティの `AverageCpuUsed` 属性の PM しきい値テンプレートが作成されます。この PM しきい値条件のしきい値条件では、`AverageCpuUsed` 属性をモニタして、CPU 平均使用率が 25 % より大きいかどうかを決定します。

```
RP/0/RP0/CPU0:Router# configure
RP/0/RP0/CPU0:Router(config)# performance-mgmt thresholds node cpu template template20
RP/0/RP0/CPU0:Router(config-threshold-cpu)# AverageCpuUsed gt 25 percent
RP/0/RP0/CPU0:Router(config-threshold-cpu)# exit
RP/0/RP0/CPU0:Router(config)# performance-mgmt apply thresholds node cpu location all
template20
RP/0/RP0/CPU0:Router# commit
```

## 正規表現によるインスタンス フィルタリングの設定

このタスクでは、1 つ以上の統計テンプレートまたはしきい値テンプレートに適用できる正規表現グループの定義について説明します。複数の正規表現インデックスを含めることもできます。正規表現グループを使用したインスタンス フィルタリングの利点は次のとおりです。

- 複数のテンプレートに適用できる同じ正規表現グループを使用できる。
- 同じ索引値を割り当てると、柔軟性を向上できる。
- 正規表現の OR 条件を適用すると、パフォーマンスを向上できる。



(注) 正規表現によるインスタンスのフィルタリングは、現在インターフェイスエンティティでのみサポートされています (Interface basic-counters、generic-counters、data-rates)。

### 設定例

次に、正規表現グループを定義する例を示します。

```
RP/0/RP0/CPU0:Router# configure
RP/0/RP0/CPU0:Router(config)# performance-mgmt regular-expression regexp
RP/0/RP0/CPU0:Router(config-perfmgmt-regex)# index 10 match
RP/0/RP0/CPU0:Router(config)# commit
```

## パフォーマンス管理：詳細

この項には、パフォーマンス管理を構成する際に役立つ追加情報が含まれています。

この表では、PM システムを構成するすべてのエンティティの各属性に関連付けられている属性と値の範囲を説明します。

表 4: 属性と値

エンティティ	属性	説明	値
<b>bgp</b>	ConnDropped	接続がドロップされた回数。	範囲は 0 ~ 4294967295 です。
	ConnEstablished	接続が確立された回数。	範囲は 0 ~ 4294967295 です。
	ErrorsReceived	接続で受信されたエラー通知の数。	範囲は 0 ~ 4294967295 です。
	ErrorsSent	接続で送信されたエラー通知の数。	範囲は 0 ~ 4294967295 です。
	InputMessages	受信されたメッセージの数。	範囲は 0 ~ 4294967295 です。
	InputUpdateMessages	受信されたアップデートメッセージの数。	範囲は 0 ~ 4294967295 です。
	OutputMessages	送信されたメッセージの数。	範囲は 0 ~ 4294967295 です。
	OutputUpdateMessages	送信されたアップデートメッセージの数。	範囲は 0 ~ 4294967295 です。

エンティティ	属性	説明	値
interface data-rates	Bandwidth	帯域幅 (kbps 単位)。	範囲は 0 ~ 4294967295 です。
	InputDataRate	入力データ レート (kbps 単位)。	範囲は 0 ~ 4294967295 です。
	InputPacketRate	入力パケット/秒。	範囲は 0 ~ 4294967295 です。
	InputPeakRate	ピーク入力データ レート。	範囲は 0 ~ 4294967295 です。
	InputPeakPkts	ピーク入力パケット レート。	範囲は 0 ~ 4294967295 です。
	OutputDataRate	出力データ レート (kbps 単位)。	範囲は 0 ~ 4294967295 です。
	OutputPacketRate	出力パケット/秒。	範囲は 0 ~ 4294967295 です。
	OutputPeakPkts	ピーク出力パケット レート。	範囲は 0 ~ 4294967295 です。
	OutputPeakRate	ピーク出力データ レート。	範囲は 0 ~ 4294967295 です。

エンティティ	属性	説明	値
<b>interface basic-counters</b>	InPackets	受信されたパケット数。	範囲は 0 ~ 4294967295 です。
	InOctets	受信されたバイト数。	範囲は 0 ~ 4294967295 です。
	OutPackets	送信されたパケット数。	範囲は 0 ~ 4294967295 です。
	OutOctets	送信されたバイト数。	範囲は 0 ~ 4294967295 です。
	InputTotalDrops	インバウンドの廃棄された適正なパケット。	範囲は 0 ~ 4294967295 です。
	InputQueueDrops	入力キューのドロップ。	範囲は 0 ~ 4294967295 です。
	InputTotalErrors	インバウンドの廃棄された不正なパケット。	範囲は 0 ~ 4294967295 です。
	OutputTotalDrops	アウトバウンドの廃棄された適正なパケット。	範囲は 0 ~ 4294967295 です。
	OutputQueueDrops	出力キューのドロップ。	範囲は 0 ~ 4294967295 です。
	OutputTotalErrors	アウトバウンドの廃棄された不正なパケット。	範囲は 0 ~ 4294967295 です。

エンティティ	属性	説明	値
<b>interface generic-counters</b>	InBroadcastPkts	受信されたブロードキャストパケット。	範囲は 0 ～ 4294967295 です。
	InMulticastPkts	受信されたマルチキャストパケット。	範囲は 0 ～ 4294967295 です。
	InOctets	受信されたバイト数。	範囲は 0 ～ 4294967295 です。
	InPackets	受信されたパケット数。	範囲は 0 ～ 4294967295 です。
	InputCRC	不正な CRC で廃棄されたインバウンドパケット。	範囲は 0 ～ 4294967295 です。
	InputFrame	インバウンドフレームエラー。	範囲は 0 ～ 4294967295 です。
	InputOverrun	入力オーバーラン。	範囲は 0 ～ 4294967295 です。
	InputQueueDrops	入力キューのドロップ。	範囲は 0 ～ 4294967295 です。
	InputTotalDrops	インバウンドの廃棄された適正なパケット。	範囲は 0 ～ 4294967295 です。
	InputTotalErrors	インバウンドの廃棄された不正なパケット。	範囲は 0 ～ 4294967295 です。
	InUcastPkts	受信されたユニキャストパケット。	範囲は 0 ～ 4294967295 です。
	InputUnknownProto	不明なプロトコルで廃棄されたインバウンドパケット。	範囲は 0 ～ 4294967295 です。
	OutBroadcastPkts	送信されたブロードキャストパケット。	範囲は 0 ～ 4294967295 です。
	OutMulticastPkts	送信されたマルチキャストパケット。	範囲は 0 ～ 4294967295 です。
	OutOctets		

エンティティ	属性	説明	値
		送信されたバイト数。	範囲は 0 ～ 4294967295 です。
	OutPackets	送信されたパケット数。	範囲は 0 ～ 4294967295 です。
	OutputTotalDrops	アウトバウンドの廃棄された適正なパケット。	範囲は 0 ～ 4294967295 です。
	OutputTotalErrors	アウトバウンドの廃棄された不正なパケット。	範囲は 0 ～ 4294967295 です。
	OutUcastPkts	送信されたユニキャストパケット。	範囲は 0 ～ 4294967295 です。
	OutputUnderrun	出力アンダーラン。	範囲は 0 ～ 4294967295 です。



エンティティ	属性	説明	値
mpls ldp	AddressMsgsRcvd	受信されたアドレスメッセージ。	範囲は 0 ~ 4294967295 です。
	AddressMsgsSent	送信されたアドレスメッセージ。	範囲は 0 ~ 4294967295 です。
	AddressWithdrawMsgsRcd	受信されたアドレスウィズドロメッセージ。	範囲は 0 ~ 4294967295 です。
	AddressWithdrawMsgsSent	送信されたアドレスウィズドロメッセージ。	範囲は 0 ~ 4294967295 です。
	InitMsgsSent	送信された初期メッセージ。	範囲は 0 ~ 4294967295 です。
	InitMsgsRcvd	受信された初期メッセージ。	範囲は 0 ~ 4294967295 です。
	KeepaliveMsgsRcvd	受信されたキープアライブメッセージ。	範囲は 0 ~ 4294967295 です。
	KeepaliveMsgsSent	送信されたキープアライブメッセージ。	範囲は 0 ~ 4294967295 です。
	LabelMappingMsgsRcvd	受信されたラベルマッピングメッセージ。	範囲は 0 ~ 4294967295 です。
	LabelMappingMsgsSent	送信されたラベルマッピングメッセージ。	範囲は 0 ~ 4294967295 です。
	LabelReleaseMsgsRcvd	受信されたラベルリリースメッセージ。	範囲は 0 ~ 4294967295 です。
	LabelReleaseMsgsSent	送信されたラベルリリースメッセージ。	範囲は 0 ~ 4294967295 です。
	LabelWithdrawMsgsRcvd	受信されたラベルウィズドロメッセージ。	範囲は 0 ~ 4294967295 です。
	LabelWithdrawMsgsSent	送信されたラベルウィズドロメッセージ。	範囲は 0 ~ 4294967295 です。

エンティティ	属性	説明	値
	NotificationMsgsRcvd	受信された通知メッセージ。	範囲は 0 ~ 4294967295 です。
	NotificationMsgsSent	送信された通知メッセージ。	範囲は 0 ~ 4294967295 です。
	TotalMsgsRcvd	受信されたメッセージの合計数。	範囲は 0 ~ 4294967295 です。
	TotalMsgsSent	送信されたメッセージの合計数。	範囲は 0 ~ 4294967295 です。
<b>node cpu</b>	AverageCPUUsed	平均 CPU 利用率。	範囲は 0 ~ 100 のパーセンテージです。
	NoProcesses	プロセス数。	範囲は 0 ~ 4294967295 です。
<b>node memory</b>	CurrMemory	現在使用中のアプリケーションメモリ (バイト単位)。	範囲は 0 ~ 4294967295 です。
	PeakMemory	ブートアップ後に使用された最大システムメモリ (MB 単位)。	範囲は 0 ~ 4194304 です。
<b>node process</b>	AverageCPUUsed	平均 CPU 利用率。	範囲は 0 ~ 100 のパーセンテージです。
	NoThreads	スレッド数。	範囲は 0 ~ 4294967295 です。
	PeakMemory	起動時以降に使用された最大ダイナミックメモリ (KB 単位)。	範囲は 0 ~ 4194304 です。

エンティティ	属性	説明	値
ospf v2protocol	InputPackets	受信されたパケットの合計数。	範囲は 0 ～ 4294967295 です。
	OutputPackets	送信されたパケットの合計数。	範囲は 0 ～ 4294967295 です。
	InputHelloPackets	受信された hello パケットの合計数。	範囲は 0 ～ 4294967295 です。
	OutputHelloPackets	送信された hello パケットの合計数。	範囲は 0 ～ 4294967295 です。
	InputDBDs	受信された DBD パケットの合計数。	範囲は 0 ～ 4294967295 です。
	InputDBDsLSA	DBD パケットで受信された LSA の数。	範囲は 0 ～ 4294967295 です。
	OutputDBDs	送信された DBD パケットの合計数。	範囲は 0 ～ 4294967295 です。
	OutputDBDsLSA	DBD パケットで送信された LSA の数。	範囲は 0 ～ 4294967295 です。
	InputLSRequests	受信された LS 要求の数。	範囲は 0 ～ 4294967295 です。
	InputLSRequestsLSA	LS 要求で受信された LSA の数。	範囲は 0 ～ 4294967295 です。
	OutputLSRequests	送信された LS 要求の数。	範囲は 0 ～ 4294967295 です。
	OutputLSRequestsLSA	LS 要求で送信された LSA の数。	範囲は 0 ～ 4294967295 です。
	InputLSAUpdates	受信された LSA アップデートの数。	範囲は 0 ～ 4294967295 です。
	InputLSAUpdatesLSA	LSA アップデートで受信された LSA の数。	範囲は 0 ～ 4294967295 です。
OutputLSAUpdates	送信された LSA アップデートの数。	範囲は 0 ～ 4294967295 です。	
OutputLSAUpdatesLSA		範囲は 0 ～ 4294967295 です。	

エンティティ	属性	説明	値
		LSAアップデートで送信されたLSAの数。	
	InputLSAAcks	受信されたLSAアクノレジメントの数。	範囲は0～4294967295です。
	InputLSAAcksLSA	LSAアクノレジメントで受信されたLSAの数。	範囲は0～4294967295です。
	OutputLSAAcks	送信されたLSAアクノレジメントの数。	範囲は0～4294967295です。
	OutputLSAAcksLSA	LSAアクノレジメントで送信されたLSAの数。	範囲は0～4294967295です。
	ChecksumErrors	チェックサムエラーで受信されたパケット数。	範囲は0～4294967295です。

エンティティ	属性	説明	値
ospf v3protocol	InputPackets	受信されたパケットの合計数。	範囲は 0 ～ 4294967295 です。
	OutputPackets	送信されたパケットの合計数。	範囲は 0 ～ 4294967295 です。
	InputHelloPackets	受信された hello パケットの合計数。	範囲は 0 ～ 4294967295 です。
	OutputHelloPackets	送信された hello パケットの合計数。	範囲は 0 ～ 4294967295 です。
	InputDBDs	受信された DBD パケットの合計数。	範囲は 0 ～ 4294967295 です。
	InputDBDsLSA	DBD パケットで受信された LSA の数。	範囲は 0 ～ 4294967295 です。
	OutputDBDs	送信された DBD パケットの合計数。	範囲は 0 ～ 4294967295 です。
	OutputDBDsLSA	DBD パケットで送信された LSA の数。	範囲は 0 ～ 4294967295 です。
	InputLSRequests	受信された LS 要求の数。	範囲は 0 ～ 4294967295 です。
	InputLSRequestsLSA	LS 要求で受信された LSA の数。	範囲は 0 ～ 4294967295 です。
	OutputLSRequests	送信された LS 要求の数。	範囲は 0 ～ 4294967295 です。
	OutputLSRequestsLSA	LS 要求で送信された LSA の数。	範囲は 0 ～ 4294967295 です。
	InputLSAUpdates	受信された LSA アップデートの数。	範囲は 0 ～ 4294967295 です。
	InputLSRequestsLSA	LS 要求で受信された LSA の数。	範囲は 0 ～ 4294967295 です。
OutputLSAUpdates	送信された LSA アップデートの数。	範囲は 0 ～ 4294967295 です。	
OutputLSAUpdatesLSA	LSA アップデートで送信された LSA の数。	範囲は 0 ～ 4294967295 です。	

エンティティ	属性	説明	値
	InputLSAAcks	受信されたLSAアクノレジメントの数。	範囲は0～4294967295です。
	InputLSAAcksLSA	LSAアクノレジメントで受信されたLSAの数。	範囲は0～4294967295です。
	OutputLSAAcks	送信されたLSAアクノレジメントの数。	範囲は0～4294967295です。
	OutputLSAAcksLSA	LSAアクノレジメントで送信されたLSAの数。	範囲は0～4294967295です。

この表では、さまざまなエンティティインスタンスに対しエンティティインスタンスモニタリングをイネーブルにするために使用されるコマンドを説明します。

表 5: エンティティインスタンスとモニタリングコマンド

エンティティ	コマンドの説明
BGP	<p>BGP エンティティインスタンスのエンティティインスタンスモニタリングを有効にするには、<b>performance-mgmt apply monitor bgp</b> コマンドを使用します。</p> <p>構文：</p> <pre style="text-align: center;"> <b>performance-mgmt</b>   <b>apply monitor</b>   <b>bgp</b>   ip-address   template-name   <b>default</b> </pre> <p>RP/0/RP0/CPU0:Router(config)# performance-mgmt apply monitor bgp 10.12.0.4 default</p>

エンティティ	コマンドの説明
インターフェイスデータレート	<p>インターフェイスデータレートのエンティティインスタンスのエンティティインスタンスモニタリングを有効にするには、<b>performance-mgmt apply monitor data-rates</b> コマンドを使用します。</p> <p>構文：</p> <pre> performance-mgmt   apply   monitor   interface   data-rates     type     interface-path-id {template-name     default} RP/0/RP0/CPU0:Router(config)# performance-mgmt apply monitor interface data-rates HundredGigE 0/0/1/0 default </pre>
インターフェイス基本カウンタ	<p>インターフェイス基本カウンタのエンティティインスタンスのエンティティインスタンスモニタリングを有効にするには、<b>performance-mgmt apply monitor interface basic-counters</b> コマンドを使用します。</p> <p>構文：</p> <pre> performance-mgmt   apply   monitor   interface   basic-counters     type     interface-path-id {template-name     default} RP/0/RP0/CPU0:Router(config)# performance-mgmt apply monitor interface basic-counters HundredGigE 0/0/1/0 default </pre>

エンティティ	コマンドの説明
インターフェイス汎用カウンタ	<p>インターフェイス汎用カウンタのエンティティ インスタンスのエンティティ インスタンス モニタリングを有効にするには、 <b>performance-mgmt apply monitor interface generic-counters</b> コマンドを使用します。</p> <p>構文：</p> <pre> performance-mgmt   apply   monitor   interface   generic-counters     type     interface-path-id {template-name     default} </pre> <p>RP/0/RP0/CPU0:Router(config)# performance-mgmt apply monitor interface generic-counters HundredGigE 0/0/1/0 default</p>
MPLS LDP	<p>MPLS LDP エンティティ インスタンスのエンティティ インスタンス モニタリングを有効にするには、 <b>performance-mgmt apply monitor mpls ldp</b> コマンドを使用します。</p> <p>構文：</p> <pre> performance-mgmt   apply monitor   mpls   ldp     ip-address {template-name     default} </pre> <p>RP/0/RP0/CPU0:Router(config)# performance-mgmt apply monitor mpls ldp 10.34.64.154 default</p>
ノード CPU	<p>ノード CPU エンティティ インスタンスのエンティティ インスタンス モニタリングを有効にするには、 <b>performance-mgmt apply monitor node cpu</b> コマンドを使用します。</p> <p>構文：</p> <pre> performance-mgmt   apply   monitor   node   cpu   location     node-id {template-name     default} </pre> <p>RP/0/RP0/CPU0:Router(config)# performance-mgmt apply monitor node cpu location 0/RP0/CPU0 default</p>



エンティティ	コマンドの説明
ノードメモリ	<p>ノードメモリ エンティティ インスタンスのエンティティ インスタンス モニタリングを有効にするには、 <b>performance-mgmt apply monitor node memory</b> コマンドを使用します。</p> <p>構文：</p> <pre> performance-mgmt   apply   monitor   node   memory   location     node-id {template-name     default} RP/0/RP0/CPU0:Router(config)# performance-mgmt apply monitor node memory location 0/RP0/CPU0 default </pre>
ノードプロセス	<p>ノードプロセス エンティティ インスタンスのエンティティ インスタンス モニタリング コレクションを有効にするには、 <b>performance-mgmt apply monitor node process</b> コマンドを使用します。</p> <p>構文：</p> <pre> performance-mgmt   apply monitor node   process   location     node-id     pid {template-name   default} RP/0/RP0/CPU0:Router(config)# performance-mgmt apply monitor node process location p 0/RP0/CPU0 275 default </pre>

