

Implementing Performance Management

Performance management (PM) on the Cisco IOS XR Software provides a framework to perform these tasks:

- Collect and export PM statistics to a TFTP server for data storage and retrieval
- Monitor the system using extensible markup language (XML) queries
- Configure threshold conditions that generate system logging messages when a threshold condition is matched.

The PM system collects data that is useful for graphing or charting system resource utilization, for capacity planning, for traffic engineering, and for trend analysis.

YANG Data Model

You can programmatically monitor the system resources using `openconfig-system.yang` OpenConfig data model. To get started with using data models, see the *Programmability Configuration Guide for Cisco NCS 540 Series Routers*.

- Prerequisites for Implementing Performance Management , on page 1
- Information About Implementing Performance Management, on page 2
- PM Functional Overview, on page 2
- PM Benefits, on page 3
- PM Statistics Collection Overview, on page 3
- How to Implement Performance Management, on page 7

Prerequisites for Implementing Performance Management

Before implementing performance management in your network operations center (NOC), ensure that these prerequisites are met:

- You must install and activate the Package Installation Envelope (PIE) for the manageability software.
- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
- You must have connectivity with a TFTP server.

Information About Implementing Performance Management

PM Functional Overview

The Performance Management (PM) frameworks consists of two major components:

PM Statistics Server

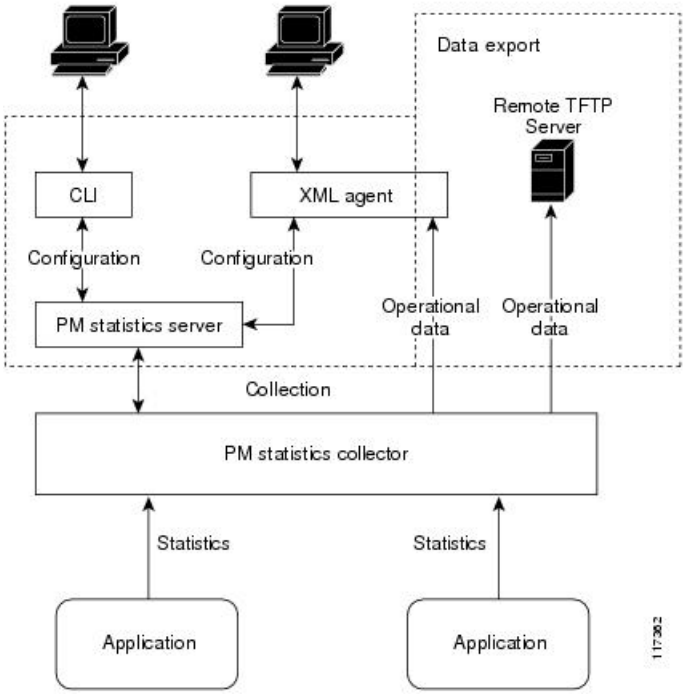
The PM statistics server is the front end for statistic collections, entity instance monitoring collections, and threshold monitoring. All PM statistic collections and threshold conditions configured through the command-line interface (CLI) or through XML schemas are processed by the PM statistics server and distributed among the PM statistics collectors.

PM Statistics Collector

The PM statistics collector collects statistics from entity instances and stores that data in memory. The memory contents are checkpointed so that information is available across process restarts. In addition, the PM statistics collector is responsible for exporting operational data to the XML agent and to the TFTP server.

Figure 1: PM Component Communications, on page 2 illustrates the relationship between the components that constitute the PM system.

Figure 1: PM Component Communications



PM Benefits

The PM system provides these benefits:

- Configurable data collection policies
- Efficient transfer of statistical data in the binary format via TFTP
- Entity instance monitoring support
- Threshold monitoring support
- Data persistency across process restarts and processor failovers

PM Statistics Collection Overview

A PM statistics collection first gathers statistics from all the attributes associated with all the instances of an entity in the PM system. It then exports the statistical data in the binary file format to a TFTP server. For example, a Multiprotocol Label Switching (MPLS) Label Distribution Protocol (LDP) statistics collection gathers statistical data from all the attributes associated with all MPLS LDP sessions on the router.

This table lists the entities and the associated instances in the PM system.

Table 1: Entity Classes and Associated Instances

Entity Classes	Instance
BGP	Neighbors or Peers
Interface Basic Counters	Interfaces
Interface Data Rates	Interfaces
Interface Generic Counters	Interfaces
MPLS LDP	LDP Sessions
Node CPU	Nodes
Node Memory	Nodes
Node Process	Processes
OSPFv2	Processes
OSPFv3	Processes



Note For a list of all attributes associated with the entities that constitute the PM system, see Table 4: Attributes and Values, on page 10.

Note Based on the interface type, the interface either supports the interface generic counters or the interface basic counters. The interfaces that support the interface basic counters do not support the interface data rates.

Binary File Format for Exporting PM Statistics

This sample describes the binary file format:

```
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
```

Binary File ID Assignments for Entity, Subentity, and StatsCounter Names

This table describes the assignment of various values and keys which is present in the binary file.

Table 2: Binary Format Values and Keys

Entity	Subentity	Key	StatsCounters
Node (1)	CPU (1)	CPU Key <Node ID>	See Table 3: Supported StatsCounters for Entities and Subentites, on page 5
	Memory (2)	Memory Key <Node ID>	
	Process (3)	Node Process Key <NodeProcessID>	
Interface (2)	Generic Counters (1)	Generic Counters Key <ifName>	
	Data Rate Counters (2)	Data Rate Counters Key <ifName>	
	Basic Counters (3)	Basic Counters Key <ifName>	
BGP (3)	Peer (1)	Peer Key <IpAddress>	

Entity	Subentity	Key	StatsCounters
MPLS (4)	Reserved (1)	—	
	Reserved (2)	—	
	LDP (4)	LDP Session Key <IpAddress>	
OSPF (5)	v2protocol (1)	Instance <process_instance>	
	v3protocol (2)	Instance <process_instance	

Note <ifName>—The length is variable. The first two bytes contain the size of the Instance ID; this is followed by the Instance ID string (that is, an Interface name).

<IpAddress>—4 bytes that contain the IP address.

<NodeProcessID>—64-bit Instance ID. The first 32 bits contain the node ID, and the second 32 bits contain the process ID.

<NodeID>—32-bit instance ID that contains the Node ID.

<process_instance>—The length is variable. The first two bytes contain the size of Instance ID followed by Instance ID string (that is, a process name).

Note The numbers in parenthesis (the numbers that are associated with each entity and subentity in Table 2: Binary Format Values and Keys, on page 4) denote the entity and subEntity IDs that are displayed in the TFTP File.

This table describes the supported statistics counters that are collected in the binary file for entities and subentities.

Table 3: Supported StatsCounters for Entities and Subentites

Entity	Subentity	StatsCounters
Node (1)	CPU (1)	NoProcesses
	Memory (2)	CurrMemory, PeakMemory
	Process (3)	PeakMemory, NoThreads
Interface (2)	Generic Counters (1)	InPackets, InOctets, OutPackets, OutOctets, InUcastPkts, InMulticastPkts, InBroadcastPkts, OutUcastPkts, OutMulticastPkts, OutBroadcastPkts, OutputTotalDrops, InputTotalDrops, InputQueueDrops, InputUnknownProto, OutputTotalErrors, OutputUnderrun, InputTotalErrors, InputCRC,InputOverrun, InputFrame

Entity	Subentity	StatsCounters
	Data Rate Counters (2)	InputDataRate, InputPacketRate, OutputDataRate, OutputPacketRate, InputPeakRate, InputPeakPkts, OutputPeakRate, OutputPeakPkts, Bandwidth
	Basic Counters	InPackets, InOctets, OutPackets, OutOctets, InputTotalDrops, InputQueueDrops,InputTotalErrors, OutputTotalErrors, OutputQueueDrops, OutputTotalErrors
BGP (3)	Peer (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

Filenaming Convention Applied to Binary Files

These filenaming convention is applied to PM statistics collections that are sent to the directory location configured on the TFTP server:

<LR_NAME>_<EntityName>_<SubentityName>_<TimeStamp>

How to Implement Performance Management

Configuring an External TFTP Server or Local Disk for PM Statistics Collection

You can export PM statistical data to an external TFTP server or dump the data to the local file system. Both the local and TFTP destinations are mutually exclusive and you can configure either one of them at a time.

Configuration Examples

This example configures an external TFTP server for PM statistics collection.

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

This example configures a local disk for PM statistics collection.

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

Configuring PM Statistics Collection Templates

PM statistics collections are configured through PM statistics collection templates. A PM statistics collection template contains the entity, the sample interval, and the number of sampling operations to be performed before exporting the data to a TFTP server. When a PM statistics collection template is enabled, the PM statistics collection gathers statistics for all attributes from all instances associated with the entity configured in the template. You can define multiple templates for any given entity; however, only one PM statistics collection template for a given entity can be enabled at a time.

Guidelines for Configuring PM Statistics Collection Templates

When creating PM statistics collection templates, follow these guidelines:

- You must configure a TFTP server resource or local dump resource if you want to export statistics data onto a remote TFTP server or local disk.
- You can define multiple templates for any given entity, but at a time you can enable only one PM statistics collection template for a given entity.
- When configuring a template, you can designate the template for the entity as the default template using the default keyword or name the template. The default template contains the following default values:
  - A sample interval of 10 minutes.
  - A sample size of five sampling operations.
- The sample interval sets the frequency of the sampling operations performed during the sampling cycle. You can configure the sample interval with the sample-interval command. The range is from 1 to 60 minutes.
- The sample size sets the number of sampling operations to be performed before exporting the data to the TFTP server. You can configure the sample size with the **sample-size** command. The range is from 1 to 60 samples.

Note Specifying a small sample interval increases CPU utilization, whereas specifying a large sample size increases memory utilization. The sample size and sample interval, therefore, may need to be adjusted to prevent system overload.

- The export cycle determines how often PM statistics collection data is exported to the TFTP server. The export cycle can be calculated by multiplying the sample interval and sample size (sample interval x sample size = export cycle).
- Once a template has been enabled, the sampling and export cycles continue until the template is disabled with the no form of the **performance-mgmt apply statistics** command.
- You must specify either a node with the **location** command or enable the PM statistic collections for all nodes using the **location all** command when enabling or disabling a PM statistic collections for the following entities:
  - Node CPU
  - Node memory
  - Node process

Configuration Example

This example shows how to create and enable a PM statistics collection template.

```
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
```

Enabling PM Entity Instance Monitoring

Entity instance monitoring gathers statistics from attributes associated with a specific entity instance. When an entity instance is enabled for monitoring, the PM system gathers statistics from only attributes associated with the specified entity instance. The PM system uses the sampling cycle that is configured in the PM statistics collection template for the entity being monitored. Entity instance monitoring, however, is a separate process from that of the PM statistics collection; therefore, it does not interfere with PM statistics collection. Furthermore, the data from entity instance monitoring collection is independent of PM statistics collection. Unlike PM statistics collection, the data from entity instance monitoring is not exported to the TFTP server. For more information about all the attributes associated with each entity instance and commands, see Performance Management: Details, on page 10.

Configuration Example

This example shows how to enable entity instance monitoring for a node CPU entity instance.

```
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
```

Configuring PM Threshold Monitoring Templates

The PM system supports the configuration of threshold conditions to monitor an attribute (or attributes) for threshold violations. Threshold conditions are configured through PM threshold monitoring templates. When a PM threshold template is enabled, the PM system monitors all instances of the attribute (or attributes) for the threshold condition configured in the template. If at end of the sample interval a threshold condition is matched, the PM system generates a system logging message for each instance that matches the threshold condition. For the list of attributes and value ranges associated with each attribute for all the entities, see Performance Management: Details, on page 10

Guidelines for Configuring PM Threshold Monitoring Templates

While you configure PM threshold monitoring templates, follow these guidelines:

- Once a template has been enabled, the threshold monitoring continues until the template is disabled with the **no** form of the **performance-mgmt apply thresholds** command.
- Only one PM threshold template for an entity can be enabled at a time.
- You must specify either a node with the **location** command or enable the PM statistic collections for all nodes using the **location all** command when enabling or disabling a PM threshold monitoring template for the following entities:
  - Node CPU
  - Node memory
  - Node process

Configuration Example

This example shows how to create and enable a PM threshold monitoring template. In this example, a PM threshold template is created for the **CurrMemory** attribute of the **node memory** entity. The threshold condition in this PM threshold condition monitors the **CurrMemory** attribute to determine whether the current memory use is greater than 50 percent.

```
Router# conf t
Router(config)# performance-mgmt thresholds node memory template template20
Router(config-threshold-cpu)# CurrMemory gt 50 percent
Router(config-threshold-cpu)# sample-interval 5
Router(config-threshold-cpu)# exit
Router(config)# performance-mgmt apply thresholds node memory location 0/RP0/CPU0 template20
Router(config)# commit
```

Configuring Instance Filtering by Regular Expression

This task explains defining a regular expression group which can be applied to one or more statistics or threshold templates. You can also include multiple regular expression indices. The benefits of instance filtering using the regular expression group is as follows.

- You can use the same regular expression group that can be applied to multiple templates.
- You can enhance flexibility by assigning the same index values.

- You can enhance the performance by applying regular expressions, which has OR conditions.

Note The Instance filtering by regular-expression is currently supported in interface entities only (Interface basic-counters, generic-counters, data-rates.

Configuration Example

This example shows how to define a regular expression group.

```
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
```

Performance Management: Details

This section contains additional information which will be useful while configuring performance management.

This table describes the attributes and value ranges associated with each attribute for all the entities that constitute the PM system.

Table 4: Attributes and Values

Entity	Attributes	Description	Values
bgp	ConnDropped	Number of times the connection was dropped.	Range is from 0 to 4294967295.
	ConnEstablished	Number of times the connection was established.	Range is from 0 to 4294967295.
	ErrorsReceived	Number of error notifications received on the connection.	Range is from 0 to 4294967295.
	ErrorsSent	Number of error notifications sent on the connection.	Range is from 0 to 4294967295.
	InputMessages	Number of messages received.	Range is from 0 to 4294967295.
	InputUpdateMessages	Number of update messages received.	Range is from 0 to 4294967295.
	OutputMessages	Number of messages sent.	Range is from 0 to 4294967295.
	OutputUpdateMessages	Number of update messages sent.	Range is from 0 to 4294967295.

Entity	Attributes	Description	Values
interface data-rates	Bandwidth	Bandwidth in kbps.	Range is from 0 to 4294967295.
	InputDataRate	Input data rate in kbps.	Range is from 0 to 4294967295.
	InputPacketRate	Input packets per second.	Range is from 0 to 4294967295.
	InputPeakRate	Peak input data rate.	Range is from 0 to 4294967295.
	InputPeakPkts	Peak input packet rate.	Range is from 0 to 4294967295.
	OutputDataRate	Output data rate in kbps.	Range is from 0 to 4294967295.
	OutputPacketRate	Output packets per second.	Range is from 0 to 4294967295.
	OutputPeakPkts	Peak output packet rate.	Range is from 0 to 4294967295.
	OutputPeakRate	Peak output data rate.	Range is from 0 to 4294967295.
interface basic-counters	InPackets	Packets received.	Range is from 0 to 4294967295.
	InOctets	Bytes received.	Range is from 0 to 4294967295.
	OutPackets	Packets sent.	Range is from 0 to 4294967295.
	OutOctets	Bytes sent.	Range is from 0 to 4294967295.
	InputTotalDrops	Inbound correct packets discarded.	Range is from 0 to 4294967295.
	InputQueueDrops	Input queue drops.	Range is from 0 to 4294967295.
	InputTotalErrors	Inbound incorrect packets discarded.	Range is from 0 to 4294967295.
	OutputTotalDrops	Outbound correct packets discarded.	Range is from 0 to 4294967295.
	OutputQueueDrops	Output queue drops.	Range is from 0 to 4294967295.
	OutputTotalErrors	Outbound incorrect packets discarded.	Range is from 0 to 4294967295.

Entity	Attributes	Description	Values
interface generic-counters	InBroadcastPkts	Broadcast packets received.	Range is from 0 to 4294967295.
	InMulticastPkts	Multicast packets received.	Range is from 0 to 4294967295.
	InOctets	Bytes received.	Range is from 0 to 4294967295.
	InPackets	Packets received.	Range is from 0 to 4294967295.
	InputCRC	Inbound packets discarded with incorrect CRC.	Range is from 0 to 4294967295.
	InputFrame	Inbound framing errors.	Range is from 0 to 4294967295.
	InputOverrun	Input overruns.	Range is from 0 to 4294967295.
	InputQueueDrops	Input queue drops.	Range is from 0 to 4294967295.
	InputTotalDrops	Inbound correct packets discarded.	Range is from 0 to 4294967295.
	InputTotalErrors	Inbound incorrect packets discarded.	Range is from 0 to 4294967295.
	InUcastPkts	Unicast packets received.	Range is from 0 to 4294967295.
	InputUnknownProto	Inbound packets discarded with unknown protocol.	Range is from 0 to 4294967295.
	OutBroadcastPkts	Broadcast packets sent.	Range is from 0 to 4294967295.
	OutMulticastPkts	Multicast packets sent.	Range is from 0 to 4294967295.
	OutOctets	Bytes sent.	Range is from 0 to 4294967295.
	OutPackets	Packets sent.	Range is from 0 to 4294967295.
	OutputTotalDrops	Outbound correct packets discarded.	Range is from 0 to 4294967295.
	OutputTotalErrors	Outbound incorrect packets discarded.	Range is from 0 to 4294967295.
	OutUcastPkts	Unicast packets sent.	Range is from 0 to 4294967295.
	OutputUnderrun	Output underruns.	Range is from 0 to 4294967295.

Entity	Attributes	Description	Values
mpls ldp	AddressMsgsRcvd	Address messages received.	Range is from 0 to 4294967295.
	AddressMsgsSent	Address messages sent.	Range is from 0 to 4294967295.
	AddressWithdrawMsgsRcd	Address withdraw messages received.	Range is from 0 to 4294967295.
	AddressWithdrawMsgsSent	Address withdraw messages sent.	Range is from 0 to 4294967295.
	InitMsgsSent	Initial messages sent.	Range is from 0 to 4294967295.
	InitMsgsRcvd	Initial messages received.	Range is from 0 to 4294967295.
	KeepaliveMsgsRcvd	Keepalive messages received.	Range is from 0 to 4294967295.
	KeepaliveMsgsSent	Keepalive messages sent.	Range is from 0 to 4294967295.
	LabelMappingMsgsRcvd	Label mapping messages received.	Range is from 0 to 4294967295.
	LabelMappingMsgsSent	Label mapping messages sent.	Range is from 0 to 4294967295.
	LabelReleaseMsgsRcvd	Label release messages received.	Range is from 0 to 4294967295.
	LabelReleaseMsgsSent	Label release messages sent.	Range is from 0 to 4294967295.
	LabelWithdrawMsgsRcvd	Label withdraw messages received.	Range is from 0 to 4294967295.
	LabelWithdrawMsgsSent	Label withdraw messages sent.	Range is from 0 to 4294967295.
	NotificationMsgsRcvd	Notification messages received.	Range is from 0 to 4294967295.
	NotificationMsgsSent	Notification messages sent.	Range is from 0 to 4294967295.
	TotalMsgsRcvd	Total messages received.	Range is from 0 to 4294967295.
	TotalMsgsSent	Total messages sent.	Range is from 0 to 4294967295.
node cpu	NoProcesses	Number of processes.	Range is from 0 to 4294967295.

Entity	Attributes	Description	Values
node memory	CurrMemory	Current application memory (in bytes) in use.	Range is from 0 to 4294967295.
	PeakMemory	Maximum system memory (in MB) used since bootup.	Range is from 0 to 4194304.
node process	NoThreads	Number of threads.	Range is from 0 to 4294967295.
	PeakMemory	Maximum dynamic memory (in KB) used since startup time.	Range is from 0 to 4194304.

Entity	Attributes	Description	Values
ospf v2protocol	InputPackets	Total number of packets received.	Range is from 0 to 4294967295.
	OutputPackets	Total number of packets sent.	Range is from 0 to 4294967295.
	InputHelloPackets	Number of Hello packets received.	Range is from 0 to 4294967295.
	OutputHelloPackets	Number of Hello packets sent.	Range is from 0 to 4294967295.
	InputDBDs	Number of DBD packets received.	Range is from 0 to 4294967295.
	InputDBDsLSA	Number of LSA received in DBD packets.	Range is from 0 to 4294967295.
	OutputDBDs	Number of DBD packets sent.	Range is from 0 to 4294967295.
	OutputDBDsLSA	Number of LSA sent in DBD packets.	Range is from 0 to 4294967295.
	InputLSRequests	Number of LS requests received.	Range is from 0 to 4294967295.
	InputLSRequestsLSA	Number of LSA received in LS requests.	Range is from 0 to 4294967295.
	OutputLSRequests	Number of LS requests sent.	Range is from 0 to 4294967295.
	OutputLSRequestsLSA	Number of LSA sent in LS requests.	Range is from 0 to 4294967295.
	InputLSAUpdates	Number of LSA updates received.	Range is from 0 to 4294967295.
	InputLSAUpdatesLSA	Number of LSA received in LSA updates.	Range is from 0 to 4294967295.
	OutputLSAUpdates	Number of LSA updates sent.	Range is from 0 to 4294967295.
	OutputLSAUpdatesLSA	Number of LSA sent in LSA updates.	Range is from 0 to 4294967295.
	InputLSAAcks	Number of LSA acknowledgements received.	Range is from 0 to 4294967295.

Entity	Attributes	Description	Values
	InputLSAAcksLSA	Number of LSA received in LSA acknowledgements.	Range is from 0 to 4294967295.
	OutputLSAAcks	Number of LSA acknowledgements sent	Range is from 0 to 4294967295.
	OutputLSAAcksLSA	Number of LSA sent in LSA acknowledgements.	Range is from 0 to 4294967295.
	ChecksumErrors	Number of packets received with checksum errors.	Range is from 0 to 4294967295.

Entity	Attributes	Description	Values
ospf v3protocol	InputPackets	Total number of packets received.	Range is from 0 to 4294967295.
	OutputPackets	Total number of packets sent.	Range is from 0 to 4294967295.
	InputHelloPackets	Number of Hello packets received.	Range is from 0 to 4294967295.
	OutputHelloPackets	Number of Hello packets sent.	Range is from 0 to 4294967295.
	InputDBDs	Number of DBD packets received.	Range is from 0 to 4294967295.
	InputDBDsLSA	Number of LSA received in DBD packets.	Range is from 0 to 4294967295.
	OutputDBDs	Number of DBD packets sent.	Range is from 0 to 4294967295.
	OutputDBDsLSA	Number of LSA sent in DBD packets.	Range is from 0 to 4294967295.
	InputLSRequests	Number of LS requests received.	Range is from 0 to 4294967295.
	InputLSRequestsLSA	Number of LSA received in LS requests.	Range is from 0 to 4294967295.
	OutputLSRequests	Number of LS requests sent.	Range is from 0 to 4294967295.
	OutputLSRequestsLSA	Number of LSA sent in LS requests.	Range is from 0 to 4294967295.
	InputLSAUpdates	Number of LSA updates received.	Range is from 0 to 4294967295.
	InputLSRequestsLSA	Number of LSA received in LS requests.	Range is from 0 to 4294967295.
	OutputLSAUpdates	Number of LSA updates sent.	Range is from 0 to 4294967295.
	OutputLSAUpdatesLSA	Number of LSA sent in LSA updates.	Range is from 0 to 4294967295.
	InputLSAAcks	Number of LSA acknowledgements received.	Range is from 0 to 4294967295.

Entity	Attributes	Description	Values
	InputLSAAcksLSA	Number of LSA received in LSA acknowledgements.	Range is from 0 to 4294967295.
	OutputLSAAcks	Number of LSA acknowledgements sent	Range is from 0 to 4294967295.
	OutputLSAAcksLSA	Number of LSA sent in LSA acknowledgements.	Range is from 0 to 4294967295.

This table describes the commands used to enable entity instance monitoring for different entity instances.

Table 5: Entity Instances and Monitoring Commands

Entity	Command Description
BGP	Use the **performance-mgmt apply monitor bgp** command to enable entity instance monitoring for a BGP entity instance. **Syntax:** **performance-mgmt** **apply monitor** **bgp** *ip-address* *template-name* | **default**} `RP/0/RP0/CPU0:Router(config)# performance-mgmt apply monitor bgp 10.12.0.4 default`

Interface Data Rates	Use the **performance-mgmt apply monitor data-rates** command to enable entity instance monitoring for an interface data rates entity instance.
	Syntax:
	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`

Entity	Command Description

Interface Basic Counters	Use the **performance-mgmt apply monitor interface basic-counters** command to enable entity instance monitoring for an interface basic counters entity instance.
	Syntax:
	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`

Interface Generic Counters	Use the **performance-mgmt apply monitor interface generic-counters** command to enable entity instance monitoring for an interface generic counters entity instance.
	Syntax:
	performance-mgmt
	apply
	monitor
	interface
	generic-counters
	type
	interface-path-id {*template-name* |
	default}
	`RP/0/RP0/CPU0:Router(config)# performance-mgmt apply monitor interface generic-counters`
	`HundredGigE 0/0/1/0 default`

MPLS LDP	Use the **performance-mgmt apply monitor mpls ldp** command to enable entity instance monitoring for an MPLS LDP entity instance. **Syntax:** **performance-mgmt** **apply monitor** **mpls** **ldp** *ip-address* {*template-name* | **default**} `RP/0/RP0/CPU0:Router(config)# performance-mgmt apply monitor mpls ldp 10.34.64.154 default`

Entity	Command Description
Node CPU	Use the **performance-mgmt apply monitor node cpu** command to enable entity instance monitoring for a node CPU entity instance. **Syntax:** **performance-mgmt** **apply** **monitor** **node** **cpu** **location** *node-id* {*template-name* | **default**} `RP/0/RP0/CPU0:Router(config)# performance-mgmt apply monitor node cpu location 0/RP0/CPU0 default`
Node Memory	Use the **performance-mgmt apply monitor node memory** command to enable entity instance monitoring for a node memory entity instance. **Syntax:** **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`
Node Process	Use the **performance-mgmt apply monitor node process** command to enable entity instance monitoring collection for a node process entity instance. **Syntax:** **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 process 275 default`

Verifying Performance Management

Use the following show commands to view the performance management details.

```
RP/0/RP0/CPU0:router show performance-mgmt monitor bgp 10.12.0.4 all-samples
RP/0/RP0/CPU0:router show performance-mgmt monitor interface data-rates gig0/0/0/0 all-samples

RP/0/RP0/CPU0:router show performance-mgmt monitor node memory location 0/RP0/CPU all-samples
Thu Apr 16 13:22:57.330 UTC
Node     : 0/RP0/CPU0
Sample No: 1   (Thu Apr 16 13:22:54 2020 UTC)
------------------------------------------------------------------------------
CurrMemory: 3852800000            PeakMemory: 0

RP/0/RP0/CPU0:router show performance-mgmt monitor node process location 0/RP0/CP0 process
 230 all-samples
```