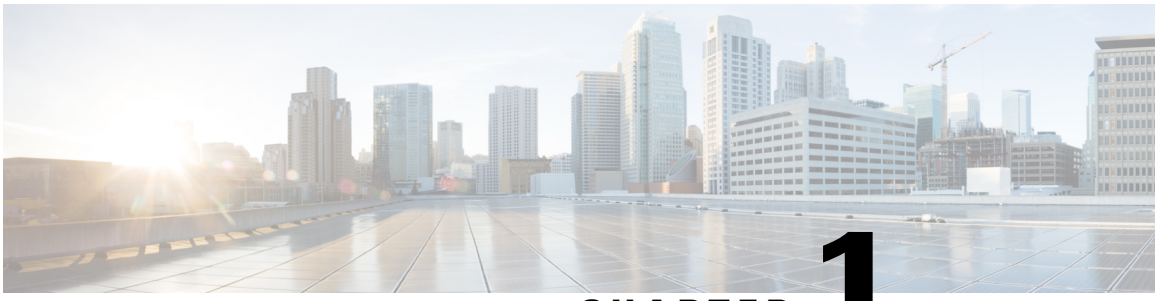
Application Visibility and Control Configuration Guide, Cisco IOS Release 15M&T

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS



CHAPTER 1

Configuring AVC to Monitor MACE Metrics

This feature is designed to analyze and measure network traffic for WAAS Express.

Application Visibility and Control (AVC) provides visibility for various applications and the network to central network management stations. MACE (Measurement, Aggregation, and Correlation Engine) provides AVC services by measuring metrics on a subset of traffic and exporting those metrics to a target. This enables the traffic to be measured and analyzed and the applications' performance to be base-lined, monitored, and troubleshot .

This feature expands on the original enhancement of the WAAS Express feature that provided support for application monitoring. Monitoring capability for Wide-Area Application Services (WAAS) Express allows the analysis and measurement of TCP-based client-server messages to provide transaction- and session-based analytics. This feature works independently of WAAS Express to provide users with application visibility.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn . An account on Cisco.com is not required.

Restrictions for Configuring AVC to Monitor MACE Metrics

MACE does not interoperate with Network Address Translation (NAT) on the ingress (LAN) interface if the **ip nat inside** command is configured on the ingress interface. However, MACE interoperates with NAT on the egress (WAN) interface if the **ip nat outside** command is configured on the egress interface.

Information about Configuring AVC to Monitor MACE Metrics

New Functionality for MACE Phase 2

Phase 2 of Measurement, Aggregation, and Correlation Engine (MACE) provides the following additional support:

- Monitoring of IPv6 flows
- MACE metrics for UDP flows.
- Two new NBAR option templates
- New option templates for class and policy information
- Use of the IPFIX protocol for flow exporters

The following collect commands can now be used to monitor IPv6 flows

- collect art response time sum
- collect art response time minimum
- collect art response time maximum
- collect art server response time sum
- collect art server response time minimum
- collect art server response time maximum
- collect art network time sum
- collect art network time minimum
- collect art network time maximum
- collect art client network time sum
- collect art client network time minimum
- collect art client network time maximum
- collect art server network time sum
- collect art server network time minimum
- collect art server network time maximum
- collect art total response time sum

- collect art total response time minimum
- collect art total response time maximum
- collect art total transaction time sum
- collect art total transaction time minimum
- collect art total transaction time maximum
- collect art count transactions
- collect art server packets
- collect art server bytes
- collect art count retrans
- collect art client packets
- collect art client bytes
- collect art count new connections
- collect art count responses
- collect art count late responses
- collect art count responses histogram
- collect art all
- collect datalink mac source address input
- collect ip dscp
- collect application name
- collect counter client bytes
- collect counter server bytes
- collect counter client packets
- collect counter server packets
- collect application http uri statistics
- collect application http host
- collect policy qos classification hierarchy
- collect policy qos queue drops
- collect time inter-packet-gap histogram

The following commands for new option templates are now supported

- option application-attributes
- option sub-application-table
- option class-qos-table
- option policy-qos-table

NetFlow Overview

NetFlow is a Cisco IOS application that provides statistics about packets that flow through a device.

NetFlow identifies packet flows for both ingress and egress IP packets. It does not involve any connection-setup protocol—either between devices or to any other networking device or end station. NetFlow does not require any external change—either to the packets themselves or to any networking device. NetFlow is completely transparent to the existing network, including end stations and application software and network devices such as LAN switches. Also, NetFlow capture and export operations are performed independently on each internetworking device; NetFlow need not be operational on each device in the network.

For more information, see the *NetFlow Configuration Guide*.

MACE Metrics

The Measurement, Aggregation, and Correlation Engine (MACE) provides the following metrics:

- MACE metrics—Metrics that are extracted or calculated by the MACE engine itself, such as the number of packets and bytes.
- ART metrics—Metrics that are extracted or calculated by the Application Response Time (ART) engine, such as network delay. These metrics are available only for TCP flows.
- WAAS metrics—Metrics that are extracted or calculated by Wide-Area Application Services (WAAS), such as Data Redundancy Elimination (DRE) input bytes. These metrics are available only when WAAS is configured and MACE is monitoring the WAAS traffic.

MACE Configuration Plane

The Measurement, Aggregation, and Correlation Engine (MACE) can be configured either through an independent and new policy-map type or as part of the Wide-Area Application Services (WAAS) policy.

The table below lists the categories of MACE configuration.

Table 1: MACE Configuration Categories

Configuration	Description
Global set of metrics	Metrics that need to be collected.
Filters	Subset of traffic for which metrics need be collected. You can configure the MACE to monitor specific traffic. The MACE uses filters to classify traffic that has to be analyzed.
Timers	Frequency with which data needs to be exported. You can configure timer values for exporting flow metrics. After the timer expires, flow metrics are exported using NetFlow Data Export Version 9 (NDE v9). This timer has a default value of 5 minutes.

Configuration	Description
NetFlow Collector's details	Details of the NetFlow Collector where data needs to be exported. You can configure information from the NetFlow Collector to export flow metrics. You can configure more than one exporter for the same set of metrics, in which metrics are exported to all NetFlow collectors.

The MACE collects the required metrics by using the metric template that contains a specific set of metric fields and exports them by using the Flexible NetFlow (FNF) infrastructure.

WAAS Express

Cisco's WAAS Express software interoperates with WAN optimization headend applications from Cisco. Cisco WAAS Express improves WAN access and use by optimizing applications, such as backup (is backup an application or a mechanism?), that require high bandwidth or are bound to a LAN.

WAAS Express helps enterprises meet the following objectives:

- Complement the Cisco WAN optimization system by adding the capability to branch routers.
- Provide branch office employees with LAN-like access to information and applications across a geographically distributed network.
- Minimize unnecessary WAN bandwidth consumption through the use of advanced compression algorithms.
- Virtualize print and other local services to branch office users.
- Improve application performance over WAN by addressing the following common issues:
    - Low data rates (constrained bandwidth)
    - Slow delivery of frames (high network latency)
    - Higher rates of packet loss (low reliability)

The Network Analysis Module (NAM) Performance Agent (PA) for WAAS Express analyzes and measures network traffic. The PA enables baselining, monitoring, and troubleshooting of application performance. The analysis and measurement of network traffic is done by the Measurement, Aggregation, and Correlation Engine (MACE). MACE performs the required measurements on a subset of traffic and exports the necessary metrics to a target.

ART Engine

The Measurement, Aggregation, and Correlation Engine (MACE) data plane forwards packets to the Application Response Time (ART) engine in the same order in which the MACE receives them. The ART engine checks every packet forwarded by the MACE.

The ART engine saves some data from each packet in its own data structures and performs the required calculations. It aggregates the flows based on the following Layer 7 (L7) information:

- Destination address
- Destination port
- Layer 4 protocol
- Segment ID
- Source address

When the export timer expires, the ART engine provides its flows and flow metrics to the MACE Exporter.

MACE Exporter

The Measurement, Aggregation, and Correlation Engine (MACE) Exporter receives the Flexible NetFlow (FNF) templates from the MACE configuration plane and builds FNF records based on these templates. It then passes the flow templates along with each record to the NetFlow infrastructure. FNF requires these templates to understand the layout of the records so that it can export the correct fields at the time of export.

The MACE Exporter allows you to configure the export time interval. The intervals 1, 2, 5, 10, and 15, in minutes, are supported. The export timer starts when the MACE is enabled. There are two ways to enable MACE: by using the MACE policy or by using the MACE along with the WAAS policy. To synchronize the export time of multiple devices that run the MACE across the network with the collector, the export timer expires when the current time modulo configured interval is zero. For instance, if a user configures a 5 minute interval at 10:07, the first export timer will expire at 10:10 (because 10:10 modulo 5 is 0) and subsequently at a gap of every 5 minutes (10:15, 10:20, and so on).

Note Modulo is the resulting remainder when one number is divided by another. For example, the modulo of 5 and 4 is 1 because 5 divided by 4 leaves a remainder of 1.

This export mechanism ensures that the time when the first export interval expires is independent from the time when the MACE policy was applied to the target. Any future update to the timeout interval causes the current timer to stop, and a new timer starts. The timer also stops when the policy is removed from the interface.

Note The MACE Exporter works on a best-effort basis. Also, MACE being a monitoring tool, the export process does execute with a high priority.

When the MACE Exporter timer expires, all engines are notified to process the metrics. After this notification, a second set of calls are sent to collect the processed metrics. The MACE Exporter receives the metrics data from various sources, aggregates them into a single FNF record, and passes it to the NetFlow component. Aggregation is done on the basis of Layer 7 keys. Application ID (Network-Based Application Recognition [NBAR]) is provided as a metric only when requested through the configuration.

How to Configure AVC to Monitor MACE Metrics

Configuring MACE for WAAS

MACE phase 2 can be invoked immediately before and after WAAS is enabled in both ingress and egress directions. This allows for measurements to be captured with no interference from any other feature. However, in the absence of WAAS, the before-WAAS and after-WAAS traffic is identical. Perform this task to enable MACE phase 2 on WAAS.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow record type mace** *name*
4. **collect art all**
5. **collect application http host**
6. **collect application http uri statistics**
7. **collect policy qos classification hierarchy**
8. **collect policy qos queue drops**
9. **collect time inter-packet-gap histogram**
10. **exit**
11. **flow exporter** *exporter-name*
12. **export-protocol ipfix**
13. **option application-attributes**
14. **option sub-application-table**
15. **option class-qos-table**
16. **option policy-qos-table**
17. **destination** *ip-address*
18. **exit**
19. **flow monitor type mace** *name*
20. **record** *record-name*
21. **exporter** *exporter-name*
22. **exit**
23. **mace monitor waas** {**all** | **optimized**} *name*
24. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow record type mace** *name* **Example:** `Device(config)# flow record type mace my-flow-record`	Configures a flow record for MACE and enters Flexible NetFlow flow record configuration mode.
Step 4	**collect art all** **Example:** `Device(config-flow-record)# collect art all`	Collects all Application Response Time (ART) metrics.
Step 5	**collect application http host** **Example:** `Device(config-flow-record)# collect http host`	Collects all Application Response Time (ART) metrics.
Step 6	**collect application http uri statistics** **Example:** `Device(config-flow-record)# collect http uri statistics`	Collects application HTTP URI statistics.
Step 7	**collect policy qos classification hierarchy** **Example:** `Device(config-flow-record)# collect policy qos classification hierarchy`	Collects the QoS policy classification hierarchy.
Step 8	**collect policy qos queue drops** **Example:** `Device(config-flow-record)# collect policy qos queue drops`	Collects the number of QoS policy queue drops.

	Command or Action	Purpose
Step 9	**collect time inter-packet-gap histogram** **Example:** `Device(config-flow-record)# collect time inter-packet-gap histogram`	Collects the inter-packet-gap time histogram.
Step 10	**exit** **Example:** `Device(config-flow-record)# exit`	Exits Flexible NetFlow flow record configuration mode.
Step 11	**flow exporter** *exporter-name* **Example:** `Device(config)# flow exporter my-flow-exporter`	Creates a Flexible NetFlow flow exporter and enters Flexible NetFlow flow exporter configuration mode.
Step 12	**export-protocol ipfix** **Example:** `Device(config-flow-exporter)# export-protocol ipfix`	Configures IPFIX as the export protocol.
Step 13	**option application-attributes** **Example:** `Device(config-flow-exporter)# option application-attributes`	Configures an option template.
Step 14	**option sub-application-table** **Example:** `Device(config-flow-exporter)# option sub-application-table`	Configures an option template.
Step 15	**option class-qos-table** **Example:** `Device(config-flow-exporter)# option class-qos-table`	Configures an option template.

	Command or Action	Purpose
Step 16	**option policy-qos-table** **Example:** `Device(config-flow-exporter)# option policy-qos-table`	Configures an option template.
Step 17	**destination** *ip-address* **Example:** `Device(config-flow-exporter)# destination 209.165.201.1`	Configures the IP address of the workstation to which you want to send the NetFlow information.
Step 18	**exit** **Example:** `Device(config-flow-exporter)# exit`	Exits Flexible NetFlow flow exporter configuration mode.
Step 19	**flow monitor type mace** *name* **Example:** `Device(config)# flow monitor type mace my-flow-monitor`	Configures a Flexible NetFlow flow monitor of type MACE and enters Flexible NetFlow flow monitor configuration mode.
Step 20	**record** *record-name* **Example:** `Device(config-flow-monitor)# record my-flow-record`	Specifies the name of a user-defined flow record that was previously configured.
Step 21	**exporter** *exporter-name* **Example:** `Device(config-flow-monitor)# exporter my-flow-exporter`	Specifies the name of a flow exporter that was previously configured.
Step 22	**exit** **Example:** `Device(config-flow-monitor)# exit`	Exits Flexible NetFlow flow monitor configuration mode.
Step 23	**mace monitor waas** {**all** | **optimized**} *name* **Example:** `Device(config)# mace monitor waas all my-flow-monitor`	Enables MACE on WAAS for a flow monitor that was previously configured.

	Command or Action	Purpose
Step 24	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring MACE for an Interface

You can enable the Cisco IOS NAM PA for WAAS Express feature on both ingress and egress interfaces so that MACE can capture and monitor traffic in both directions. After enabling MACE in one direction, the same policy is internally configured in the other direction as well. Perform this task to enable MACE on an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow record type mace** *name*
4. **collect art all**
5. **collect application name**
6. **collect counter client bytes**
7. **collect counter server bytes**
8. **collect counter client packets**
9. **collect counter client packets**
10. **collect application http host**
11. **collect application http uri statistics**
12. **collect policy qos classification hierarchy**
13. **collect policy qos queue drops**
14. **collect time inter-packet-gap histogram**
15. **exit**
16. **flow exporter** *exporter-name*
17. **export-protocol ipfix**
18. **option application-attributes**
19. **option sub-application-table**
20. **option class-qos-table**
21. **option policy-qos-table**
22. **destination** *ip-address*
23. **exit**
24. **flow monitor type mace** *name*
25. **record** *record-name*
26. **exporter** *exporter-name*
27. **exit**
28. **class-map type waas** *class-map-name*
29. **exit**
30. **policy-map type mace** *name*
31. **class** *name*
32. **flow monitor** *monitor-name*
33. **exit**
34. **exit**
35. **interface** *type number* [*name-tag*]
36. **mace enable**
37. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow record type mace** *name* **Example:** `Device(config)# flow record type mace my-flow-record`	Configures a flow record for MACE and enters Flexible NetFlow flow record configuration mode.
Step 4	**collect art all** **Example:** `Device(config-flow-record)# collect art all`	Collects all Application Response Time (ART) metrics.
Step 5	**collect application name** **Example:** `Device(config-flow-record)# collect application name`	Collects the application name.
Step 6	**collect counter client bytes** **Example:** `Device(config-flow-record)# collect counter client bytes`	Collects the total number of bytes from the client.
Step 7	**collect counter server bytes** **Example:** `Device(config-flow-record)# collect counter server bytes`	Collects the total number of bytes from the server.
Step 8	**collect counter client packets** **Example:** `Device(config-flow-record)# collect counter client packets`	Collects the total number of bytes from the server.

	Command or Action	Purpose
Step 9	**collect counter client packets** **Example:** `Device(config-flow-record)# collect counter server packets`	Collects the total number of packets from the server.
Step 10	**collect application http host** **Example:** `Device(config-flow-record)# collect http host`	Collects all Application Response Time (ART) metrics.
Step 11	**collect application http uri statistics** **Example:** `Device(config-flow-record)# collect http uri statistics`	Collects application HTTP URI statistics.
Step 12	**collect policy qos classification hierarchy** **Example:** `Device(config-flow-record)# collect policy qos classification hierarchy`	Collects the QoS policy classification hierarchy.
Step 13	**collect policy qos queue drops** **Example:** `Device(config-flow-record)# collect policy qos queue drops`	Collects the number of QoS policy queue drops.
Step 14	**collect time inter-packet-gap histogram** **Example:** `Device(config-flow-record)# collect time inter-packet-gap histogram`	Collects the inter-packet-gap time histogram.
Step 15	**exit** **Example:** `Device(config-flow-record)# exit`	Exits Flexible NetFlow flow record configuration mode.
Step 16	**flow exporter** *exporter-name* **Example:** `Device(config)# flow exporter my-flow-exporter`	Creates an FNF flow exporter and enters Flexible NetFlow flow exporter configuration mode.

	Command or Action	Purpose
Step 17	**export-protocol ipfix** **Example:** `Device(config-flow-exporter)# export-protocol ipfix`	Configures IPFIX as the export protocol.
Step 18	**option application-attributes** **Example:** `Device(config-flow-exporter)# option application-attributes`	Configures an option template.
Step 19	**option sub-application-table** **Example:** `Device(config-flow-exporter)# option sub-application-table`	Configures an option template.
Step 20	**option class-qos-table** **Example:** `Device(config-flow-exporter)# option class-qos-table`	Configures an option template.
Step 21	**option policy-qos-table** **Example:** `Device(config-flow-exporter)# option policy-qos-table`	Configures an option template.
Step 22	**destination** *ip-address* **Example:** `Device(config-flow-exporter)# destination 209.165.201.1`	Configures the IP address of the workstation to which you want to send the NetFlow information.
Step 23	**exit** **Example:** `Device(config-flow-exporter)# exit`	Exits Flexible NetFlow flow exporter configuration mode.

	Command or Action	Purpose
Step 24	**flow monitor type mace** *name* **Example:** `Device(config)# flow monitor type mace my-flow-monitor`	Configures an FNF flow monitor of type MACE and enters Flexible NetFlow flow monitor configuration mode.
Step 25	**record** *record-name* **Example:** `Device(config-flow-monitor)# record my-flow-record`	Specifies the name of a user-defined flow record that was previously configured.
Step 26	**exporter** *exporter-name* **Example:** `Device(config-flow-monitor)# exporter my-flow-exporter`	Specifies the name of a flow exporter that was previously configured.
Step 27	**exit** **Example:** `Device(config-flow-monitor)# exit`	Exits Flexible NetFlow flow monitor configuration mode.
Step 28	**class-map type waas** *class-map-name* **Example:** `Device(config)# class-map type waas my-waas-class`	Configures a WAAS Express class map and enters class map configuration mode.
Step 29	**exit** **Example:** `Device(config-cmap)# exit`	Exits class-map configuration mode.
Step 30	**policy-map type mace** *name* **Example:** `Device(config)# policy-map type mace mace_global`	Configures a MACE policy map and enters policy-map configuration mode.
Step 31	**class** *name* **Example:** `Device(config-pmap)# class my-waas-class`	Configures a class name and enters policy-map class configuration mode.

	Command or Action	Purpose
Step 32	**flow monitor** *monitor-name* **Example:** `Device(config-pmap-c)# flow monitor my-flow-monitor`	Configures a flow monitor name.
Step 33	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits policy-map class configuration mode.
Step 34	**exit** **Example:** `Device(config-pmap)# exit`	Exits policy-map configuration mode.
Step 35	**interface** *type number* [*name-tag*] **Example:** `Device(config)# interface ethernet0/0`	Configures an interface type and enters interface configuration mode.
Step 36	**mace enable** **Example:** `Device(config-if)# mace enable`	Applies the global MACE policy on an interface.
Step 37	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Flexible NetFlow commands	*Cisco IOS Flexible NetFlow Command Reference*
NetFlow configuration tasks	*Cisco IOS NetFow Configuration Guide*

Related Topic	Document Title
WAN configuration tasks	• *Wide-Area Networking Configuration Guide: Frame Relay* • *Wide-Area Networking Configuration Guide: Layer 2 Services* • *Wide-Area Networking Configuration Guide: SMDS and X.25 and LAPB* • *Wide-Area Networking Configuration Guide: Wide-Area Application Services*
WAN commands	*Cisco IOS Wide-Area Networking Command Reference*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

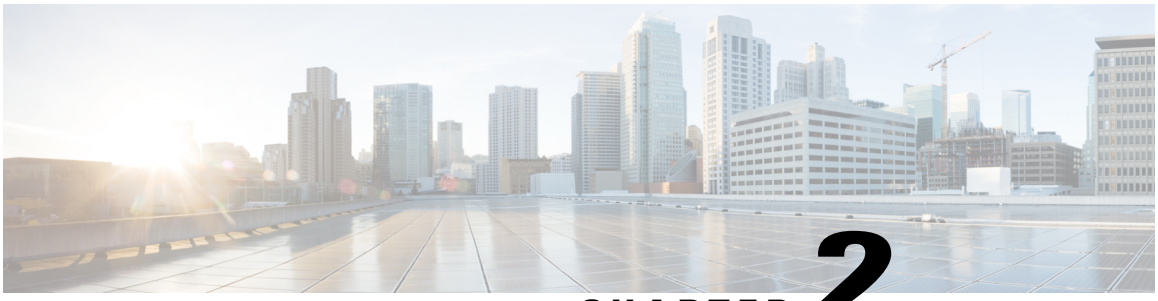
Feature Information for Configuring AVC to Monitor MACE Metrics

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to . An account on Cisco.com is not required.

Table 2: Feature Information for MACE Phase 2

Feature Name	Releases	Feature Information
MACE Phase 2	15.1(4)M2	This feature is provides support for IPv6 flows, MACE metrics for UDP flows, two new NBAR option templates, new option templates for class and policy information, and the use of IPFIX for flow exporters. The following commands were introduced or modified: **collect application http host, collect application http uri statistics, collect policy qos classification hierarchy, collect policy qos queue drops, collect time inter-packet-gap histogram, export-protocol ipfix, option application-attributes, option sub-application-table, option class-qos-table,and option policy-qos-table**.



CHAPTER 2

Easy Performance Monitor

The Easy Performance Monitor chapter describes how to configure Easy Performance Monitor (ezPM) for Application Visibility and Control (AVC).

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About Easy Performance Monitor

Easy Performance Monitor

The Easy Performance Monitor (Easy perf-mon or ezPM) feature provides an express method of provisioning monitors. This new mechanism adds functionality and does not affect the existing methods for provisioning monitors.

EzPM does not provide the full flexibility of the traditional perf-mon configuration model but provides 'profiles' that represent typical deployment scenarios. On selecting a profile and specifying a few parameters, ezPM provides the remaining provisioning information.

Profile

A profile is a pre-defined set of traffic monitors that can be enabled or disabled for a context. A profile also includes an exporter template. The following profiles are available for configuration:

- Application Experience
- Application Statistics

Traffic Monitor

A traffic monitor is a pre-defined performance monitor configuration that is used for collecting a set of metrics on a specific traffic. This is based on typical configuration that is recommended.

Each traffic monitor defines the parameters that can be modified. During configuring the traffic monitor, the CLI displays the keywords based on the parameters that can be modified.

Traffic Monitors for Application Experience Profile

For the application experience profile, the following objects are defined for both IPv4 and IPv6, and optionally for ingress/egress traffic:

- Perf-mon flow record

  Match fields (including AOR indication if/where applicable)
  Collect fields

- Perf-mon flow monitor

  Cache type
  Cache size (factor that is adjusted per platform and configured by using the **traffic-monitor application-response-time cache-size** command)
  Cache timeout (synchronized)
  History size

- Classification - definition of traffic that is monitored by the perf-mon flow monitor

Traffic Monitors for Application Statistics Profile

For an Application Statistics profile, the following traffic monitors are available for configuration:

- application-client-server-stats
- application-stats

The application statistics profile provides only application statistics and not performance statistics. The monitors operate on both IPv4 and IPv6 traffic. You can monitor either **application-client-server-stats** or **application-stats** as the **application-client-server-stats** monitor provides the same information as that of **application-stats** along with additional information.

Context

A context represents a performance monitor policy map that is attached to an interface in ingress and egress directions. A context contains the information about the traffic-monitor that has to be enabled. When a context is attached to an interface, two policy-maps are created, one each in ingress and egress directions. Depending on the direction specified in the traffic monitor, the policy-maps are attached in that direction and the traffic is monitored. You can modify the context to override pre-defined directions.

You can create multiple contexts based on a single profile with different traffic monitors, different exporters, and different parameters for every selected traffic monitor.

An ezPM context can be attached to multiple interfaces. Only one context can be attached to an interface. The context can be from any currently available profile, such as Application Experience or Application Statistics.

You can modify the ezPM context only when the context is not attached to an interface. To detach the context from an interface, use the **no performance monitor context** *context-name* command.

How to Configure Easy Performance Monitor

Configuring Easy Performance Monitor

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **performance monitor context** *context-name* **profile** *profile-name*
4. **exporter destination** {*hostname* | *ipaddress*} **source interface** *interface-type number* [**port** *port-value* **transport udp vrf** *vrf-name*]
5. (Optional) Repeat Step 4 to configure upto 3 exporters.
6. **traffic monitor** {**application-response-time** | **application-traffic-stats** | **conversation-traffic-stats** | **media** [**egress** | **ingress**] | **url**} [**cache-size** *max-entries*] [**cache-type** [**normal** | **synchronized**]] [[**class-and** | **class-replace**] *class-name*] [**ipv4** | **ipv6**]
7. Repeat Step 6 to configure additional traffic monitor parameters.
8. **exit**
9. **interface** *interface-type number*
10. **performance monitor context** *context-name*
11. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**performance monitor context** *context-name* **profile** *profile-name* **Example:** `Device(config)# performance monitor context perf-mon-test profile application-experience`	Enters performance monitor configuration mode, creates a context with application-experience profile.

	Command or Action	Purpose
Step 4	**exporter destination** {*hostname* | *ipaddress*} **source interface** *interface-type number* [**port** *port-value* **transport udp vrf** *vrf-name*] **Example:** `Device(config-perf-mon)# exporter destination 10.1.1.1 source interface GigabitEthernet0/0 port 1 transport udp vrf vpn1`	Attaches the exporter to the context and configures exporter parameters.
Step 5	(Optional) Repeat Step 4 to configure upto 3 exporters.	—
Step 6	**traffic monitor** {**application-response-time** | **application-traffic-stats** | **conversation-traffic-stats** | **media** [**egress** | **ingress**] | **url**} [**cache-size** *max-entries*] [**cache-type** [**normal** | **synchronized**]] [[**class-and** | **class-replace**] *class-name*] [**ipv4** | **ipv6**] **Example:** `Device(config-perf-mon)# traffic monitor media egress cache-size 70 class-and cln ipv6`	Configures the traffic monitor to monitor the specified metrics. **Note** The **class-and** and **class-replace** keywords are not available for application-statistics profile.
Step 7	Repeat Step 6 to configure additional traffic monitor parameters.	—
Step 8	**exit** **Example:** `Device(config-perf-mon)# exit`	Exits performance monitor configuration mode and enters global configuration mode.
Step 9	**interface** *interface-type number* **Example:** `Device(config)# interface gigabitethernet 0/1`	Enters interface configuration mode.
Step 10	**performance monitor context** *context-name* **Example:** `Device(config-if)# performance monitor context perf-mon-test`	Configures the specified performance monitor context on the interface.
Step 11	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.

Verifying Easy Performance Monitor Configuration

Verifying Easy Performance Monitor Configuration

Table 3: Verifying Easy Performance Monitor Configuration

Command	Description
show running-configuration performance-monitor context	Displays ezPM configuration of the specified context. If a context name is not specified, all contexts are displayed.

Verifying Easy Performance Monitor Profile

Table 4: Verifying Easy Performance Monitor Profile

Command	Description
show performance monitor profile *profile-name*	Displays profile information of all traffic monitors available and the active status of traffic monitors.
show performance monitor profile *profile-name* **traffic-monitor** *traffic-monitor-name*	Displays profile information such as records, monitors, and default classification for a specific traffic monitor.

Verifying Easy Performance Monitor Context

Table 5: Verifying Easy Performance Monitor Context

Command	Description
show performance-monitor context *context-name* **configuration**	Displays all configuration of the specified context. This command can be used to convert the auto configuration to the traditional configuration.
show performance-monitor context *context-name* **summary**	Displays the information about the enabled traffic monitors and the interfaces to which they are attached.

show performance-monitor context *context-name* **interface** *interface-name*	Displays the information about the performance monitor interface. The output is the same as the **show policy-map type performance-monitor interface** command.
show performance-monitor context *context-name* **traffic-monitor** *traffic-monitor-name* **cache**	Displays performance monitor cache information. The output is similar to performance-monitor cache.
show performance-monitor context *context-name* **traffic-monitor** *traffic-monitor-name* **history**	Displays performance monitor history information. The output is similar to performance-monitor history.
show performance-monitor context *context-name* **traffic-monitor** *traffic-monitor-name* **aggregate**	Displays performance monitor aggregate information. The output is similar to performance-monitor aggregate.
show performance-monitor context *context-name* **exporter**	Displays the operational information about the exporters attached to the specified context.

Debugging Easy Performance Monitor Configuration

Debugging Context

Table 6: Debugging Context

Command	Description
debug performance monitor context info	Debug context information.
debug performance monitor context error	Debug context error.

Debugging Profile

Table 7: Debugging Profile

Command	Description
debug performance monitor profile info	Debug profile information.
debug performance monitor profile error	Debug profile error.

Troubleshooting Easy Performance Monitor

Troubleshooting Easy Performance Monitor

- If no records are exported:

  For application-traffic-stats and media monitors, use the **show performance monitor context** *context-name* **traffic-monitor** *traffic-monitor-name* **cache** command to check if any records exist in the cache after the traffic is sent. The output displays any records that changed from current entries to flow-aged status on reaching the cache timeout. To see the records in the cache, use the show performance monitor cache detail command.

  For application-response-time, conversation-traffic-stats, and url monitors, use the **show performance monitor context** *context-name* **traffic-monitor** *traffic-monitor-name* **history** command to view the list of entries.

- If no records are added to the cache, you can use the **show performance monitor context** *context-name* **interface** *interface-name* command to check if the class-maps that originated automatically by ezPM are receiving the packets.
- Use the **show performance monitor context** *context-name* **configuration** command to verify the configuration that originated from ezPM. See Example: Verifying the Complete Configuration for a Performance Monitor Context

Configuration Examples for Configuring Easy Performance Monitor

Example: Configuring a Performance Monitor Context with Default ART, Media, and URL Traffic Monitors

The following example shows how to configure a performance monitor context to monitor the traffic metrics for ART, media, and URL:

```
Device# configure terminal
Device(config)# performance monitor context perf-mon-test profile application-experience
Device(config-perf-mon)# exporter destination 10.10.1.1 source interface GigabitEthernet0/0
 port 15 transport udp vrf in-vrf
Device(config-perf-mon)# traffic-monitor application-response-time
Device (config-perf-mon)# traffic-monitor media
Device(config-perf-mon)# traffic-monitor url
Device(config-perf-mon)# exit
Device(config)# interface gigabitethernet 0/1
Device(config-if)# performance monitor context perf-mon-test
Device(config-if)# exit
```

Example: Configuring a Performance Monitor Context With Media Traffic Monitor for Ipv6 Ingress and IPv4 Egress

The following example shows how to configure a performance monitor context with traffic monitor enabling the media metrics for ipv6 traffic in ingress and ipv4 traffic in egress:

```
Device# configure terminal
Device(config)# performance monitor context perf-mon-test profile application-experience
Device(config-perf-mon)# exporter destination 10.10.1.1 source interface GigabitEthernet0/0
 port 15 transport udp vrf in-vrf
Device(config-perf-mon)# traffic-monitor media ingress ipv6
Device(config-perf-mon)# traffic-monitor media egress ipv4
Device(config-perf-mon)# exit
Device(config)# interface gigabitethernet 0/1
Device(config-if)# performance monitor context perf-mon-test
Device(config-if)# exit
```

Example: Configuring a Performance Monitor Context on Multiple Interfaces

The following example shows how to configure a performance monitor context on multiple interfaces:

```
Device# configure terminal
Device(config)# interface gigabitethernet 0/1
Device(config-if)# performance monitor context perf-mon-test
Device(config-if)# exit
Device(config)# interface gigabitethernet 0/2
Device(config-if)# performance monitor context perf-mon-test
Device(config-if)# exit
```

Example: Modifying Cache Size Entries for a Traffic Monitor

The following example shows how to modifying cache size entries for a traffic monitor:

```
Device# configure terminal
Device(config)# performance monitor context perf-mon-test profile application-experience
Device(config-perf-mon)# exporter destination 10.10.1.1 source interface GigabitEthernet0/0
 port 15 transport udp vrf in-vrf
Device(config-perf-mon)# traffic-monitor application-response-time cache-size 1000
Device(config-perf-mon)# exit
```

Example: Verifying the Complete Configuration for a Performance Monitor Context

The following example shows the complete underlying configuration of a performance monitor context that uses all traffic monitors. This configuration demonstrates how ezPM builds the configuration by applying the Application Experience profile definition to the context.

```
Device# show performance monitor context reference configuration

!================================================================================
!                 Equivalent Configuration of Context reference                 !
!================================================================================
!Exporters
```

```
!==========
!
!
flow exporter reference-1
description performance monitor context reference exporter
destination 1.1.1.1
source GigabitEthernet0/1
transport udp 3333
export-protocol ipfix
template data timeout 300
option c3pl-class-table timeout 300
option c3pl-policy-table timeout 300
option interface-table timeout 300
option vrf-table timeout 300
option sampler-table timeout 300
option application-table timeout 300
option application-attributes timeout 300
option sub-application-table timeout 300
!
!Access Lists
!=============
ip access-list extended reference-conv_ipv4_tcp
permit tcp any any
!
ipv6 access-list reference-conv_ipv6_tcp
permit tcp any any
!
ip access-list extended reference-conv_ipv4_udp
permit udp any any
!
ipv6 access-list reference-conv_ipv6_udp
permit udp any any
!
ip access-list extended reference-art_ipv4_tcp
permit tcp any any
!
ipv6 access-list reference-art_ipv6_tcp
permit tcp any any
!
ip access-list extended reference-media_ipv4_udp
permit udp any any
!
ipv6 access-list reference-media_ipv6_udp
permit udp any any
!
ip access-list extended reference-url_ipv4_tcp
permit tcp any any
!
ipv6 access-list reference-url_ipv6_tcp
permit tcp any any
!
!Class-maps
!===========
class-map match-any reference-app_ts
match protocol dns
match protocol dht
!
class-map match-any reference-conv_ts_ipv4
match access-group name reference-conv_ipv4_tcp
match access-group name reference-conv_ipv4_udp
!
class-map match-any reference-conv_ts_ipv6
match access-group name reference-conv_ipv6_tcp
match access-group name reference-conv_ipv6_udp
!
class-map match-all reference-art_ipv4
match access-group name reference-art_ipv4_tcp
!
class-map match-all reference-art_ipv6
match access-group name reference-art_ipv6_tcp
!
class-map match-any reference-media_app
match protocol telepresence-media
```

```
match protocol rtp
!
class-map match-all reference-media_ipv4_in
match access-group name reference-media_ipv4_udp
match class-map reference-media_app
!
class-map match-all reference-media_ipv4_out
match access-group name reference-media_ipv4_udp
match class-map reference-media_app
!
class-map match-all reference-media_ipv6_in
match access-group name reference-media_ipv6_udp
match class-map reference-media_app
!
class-map match-all reference-media_ipv6_out
match access-group name reference-media_ipv6_udp
match class-map reference-media_app
!
class-map match-any reference-url_app
match protocol napster
match protocol gotomypc
match protocol yahoo-messenger
match protocol tunnel-http
match protocol baidu-movie
match protocol flashmyspace
match protocol directconnect
match protocol audio-over-http
match protocol skype
match protocol video-over-http
match protocol pando
match protocol flashyahoo
match protocol msn-messenger
match protocol flash-video
match protocol webthunder
match protocol vnc-http
match protocol activesync
match protocol irc
match protocol realmedia
match protocol gmail
match protocol google-earth
match protocol gnutella
match protocol rtmpt
match protocol http
match protocol ms-update
match protocol rtsp
match protocol http-alt
match protocol share-point
match protocol binary-over-http
match protocol ms-sms
match protocol megavideo
!
class-map match-all reference-url_ipv4
match access-group name reference-url_ipv4_tcp
match class-map reference-url_app
!
class-map match-all reference-url_ipv6
match access-group name reference-url_ipv6_tcp
match class-map reference-url_app
!
class-map match-all reference-art_url_ipv4
match class-map reference-art_ipv4
match class-map reference-url_ipv4
!
class-map match-all reference-art_url_ipv6
match class-map reference-art_ipv6
match class-map reference-url_ipv6
!
!Samplers
!=========
!Records and Monitors
!=====================
!
flow record type performance-monitor reference-app_ts_in
```

```
description ezPM record
match routing vrf input
match ipv4 version
match ipv4 protocol
match interface input
match flow direction
match application name account-on-resolution
collect ipv4 dscp
collect interface output
collect counter bytes long
collect counter packets
collect timestamp sys-uptime first
collect timestamp sys-uptime last
collect connection new-connections
collect connection sum-duration
!
!
flow monitor type performance-monitor reference-app_ts_in
record reference-app_ts_in
exporter reference-1
cache entries 1000
cache timeout synchronized 60
history size 0 timeout 0
!
!
flow record type performance-monitor reference-app_ts_out
description ezPM record
match routing vrf input
match ipv4 version
match ipv4 protocol
match interface output
match flow direction
match application name account-on-resolution
collect ipv4 dscp
collect interface input
collect counter bytes long
collect counter packets
collect timestamp sys-uptime first
collect timestamp sys-uptime last
collect connection new-connections
collect connection sum-duration
!
!
flow monitor type performance-monitor reference-app_ts_out
record reference-app_ts_out
exporter reference-1
cache entries 1000
cache timeout synchronized 60
history size 0 timeout 0
!
!
flow record type performance-monitor reference-conv_ts_ipv4
description ezPM record
match routing vrf input
match ipv4 protocol
match application name account-on-resolution
match connection client ipv4 address
match connection server ipv4 address
match connection server transport port
collect ipv4 dscp
collect ipv4 ttl
collect interface input
collect interface output
collect timestamp sys-uptime first
collect timestamp sys-uptime last
collect connection new-connections
collect connection sum-duration
collect connection server counter bytes long
collect connection server counter packets long
collect connection client counter bytes long
collect connection client counter packets long
!
!
```

```
flow monitor type performance-monitor reference-conv_ts_ipv4
record reference-conv_ts_ipv4
exporter reference-1
cache entries 9375
cache timeout synchronized 60 export-spread 15
history size 1
!
!
flow record type performance-monitor reference-conv_ts_ipv6
description ezPM record
match routing vrf input
match ipv6 protocol
match application name account-on-resolution
match connection client ipv6 address
match connection server transport port
match connection server ipv6 address
collect ipv6 dscp
collect ipv6 hop-limit
collect interface input
collect interface output
collect timestamp sys-uptime first
collect timestamp sys-uptime last
collect connection new-connections
collect connection sum-duration
collect connection server counter bytes long
collect connection server counter packets long
collect connection client counter bytes long
collect connection client counter packets long
!
!
flow monitor type performance-monitor reference-conv_ts_ipv6
record reference-conv_ts_ipv6
exporter reference-1
cache entries 9375
cache timeout synchronized 60 export-spread 15
history size 1
!
!
flow record type performance-monitor reference-art_ipv4
description ezPM record
match routing vrf input
match ipv4 protocol
match application name account-on-resolution
match connection client ipv4 address
match connection server ipv4 address
match connection server transport port
collect ipv4 dscp
collect ipv4 ttl
collect interface input
collect interface output
collect timestamp sys-uptime first
collect timestamp sys-uptime last
collect connection new-connections
collect connection sum-duration
collect connection delay response to-server sum
collect connection server counter responses
collect connection delay response to-server histogram late
collect connection delay network to-server sum
collect connection delay network to-client sum
collect connection client counter packets retransmitted
collect connection delay network client-to-server sum
collect connection delay application sum
collect connection delay application max
collect connection delay response client-to-server sum
collect connection transaction duration sum
collect connection transaction counter complete
collect connection server counter bytes long
collect connection server counter packets long
collect connection client counter bytes long
collect connection client counter packets long
!
!
flow monitor type performance-monitor reference-art_ipv4
```

```
record reference-art_ipv4
exporter reference-1
cache entries 10750
cache timeout synchronized 60 export-spread 15
history size 1
!
!
flow record type performance-monitor reference-art_ipv6
description ezPM record
match routing vrf input
match ipv6 protocol
match application name account-on-resolution
match connection client ipv6 address
match connection server transport port
match connection server ipv6 address
collect ipv6 dscp
collect ipv6 hop-limit
collect interface input
collect interface output
collect timestamp sys-uptime first
collect timestamp sys-uptime last
collect connection new-connections
collect connection sum-duration
collect connection delay response to-server sum
collect connection server counter responses
collect connection delay response to-server histogram late
collect connection delay network to-server sum
collect connection delay network to-client sum
collect connection client counter packets retransmitted
collect connection delay network client-to-server sum
collect connection delay application sum
collect connection delay application max
collect connection delay response client-to-server sum
collect connection transaction duration sum
collect connection transaction counter complete
collect connection server counter bytes long
collect connection server counter packets long
collect connection client counter bytes long
collect connection client counter packets long
!
!
flow monitor type performance-monitor reference-art_ipv6
record reference-art_ipv6
exporter reference-1
cache entries 10750
cache timeout synchronized 60 export-spread 15
history size 1
!
!
flow record type performance-monitor reference-media_ipv4_in
description ezPM record
match routing vrf input
match ipv4 protocol
match ipv4 source address
match ipv4 destination address
match transport source-port
match transport destination-port
match transport rtp ssrc
match interface input
collect ipv4 dscp
collect ipv4 ttl
collect transport packets lost counter
collect transport rtp jitter maximum
collect interface output
collect counter bytes long
collect counter packets
collect timestamp sys-uptime first
collect timestamp sys-uptime last
collect application name
collect connection new-connections
collect transport rtp payload-type
collect transport rtp jitter mean sum
!
```

```
!
flow monitor type performance-monitor reference-media_ipv4_in
record reference-media_ipv4_in
exporter reference-1
cache entries 4000
cache timeout synchronized 60 export-spread 15
history size 10
!
!
flow record type performance-monitor reference-media_ipv6_in
description ezPM record
match routing vrf input
match ipv6 protocol
match ipv6 source address
match ipv6 destination address
match transport source-port
match transport destination-port
match transport rtp ssrc
match interface input
collect ipv6 dscp
collect ipv6 hop-limit
collect transport packets lost counter
collect transport rtp jitter maximum
collect interface output
collect counter bytes long
collect counter packets
collect timestamp sys-uptime first
collect timestamp sys-uptime last
collect application name
collect connection new-connections
collect transport rtp payload-type
collect transport rtp jitter mean sum
!
!
flow monitor type performance-monitor reference-media_ipv6_in
record reference-media_ipv6_in
exporter reference-1
cache entries 4000
cache timeout synchronized 60 export-spread 15
history size 10
!
!
flow record type performance-monitor reference-media_ipv4_out
description ezPM record
match routing vrf input
match ipv4 protocol
match ipv4 source address
match ipv4 destination address
match transport source-port
match transport destination-port
match transport rtp ssrc
match interface output
collect ipv4 dscp
collect ipv4 ttl
collect transport packets lost counter
collect transport rtp jitter maximum
collect interface input
collect counter bytes long
collect counter packets
collect timestamp sys-uptime first
collect timestamp sys-uptime last
collect application name
collect connection new-connections
collect transport rtp payload-type
collect transport rtp jitter mean sum
!
!
flow monitor type performance-monitor reference-media_ipv4_out
record reference-media_ipv4_out
exporter reference-1
cache entries 4000
cache timeout synchronized 60 export-spread 15
history size 10
```

```
!
!
flow record type performance-monitor reference-media_ipv6_out
description ezPM record
match routing vrf input
match ipv6 protocol
match ipv6 source address
match ipv6 destination address
match transport source-port
match transport destination-port
match transport rtp ssrc
match interface output
collect ipv6 dscp
collect ipv6 hop-limit
collect transport packets lost counter
collect transport rtp jitter maximum
collect interface input
collect counter bytes long
collect counter packets
collect timestamp sys-uptime first
collect timestamp sys-uptime last
collect application name
collect connection new-connections
collect transport rtp payload-type
collect transport rtp jitter mean sum
!
!
flow monitor type performance-monitor reference-media_ipv6_out
record reference-media_ipv6_out
exporter reference-1
cache entries 4000
cache timeout synchronized 60 export-spread 15
history size 10
!
!
flow record type performance-monitor reference-url_ipv4
description ezPM record
match routing vrf input
match ipv4 protocol
match application name account-on-resolution
match connection client ipv4 address
match connection server ipv4 address
match connection server transport port
collect ipv4 dscp
collect ipv4 ttl
collect interface input
collect interface output
collect timestamp sys-uptime first
collect timestamp sys-uptime last
collect connection new-connections
collect connection sum-duration
collect application http uri statistics
collect connection delay response to-server sum
collect connection server counter responses
collect connection delay response to-server histogram late
collect connection delay network to-server sum
collect connection delay network to-client sum
collect connection client counter packets retransmitted
collect connection delay network client-to-server sum
collect connection delay application sum
collect connection delay application max
collect connection delay response client-to-server sum
collect connection transaction duration sum
collect connection transaction counter complete
collect connection server counter bytes long
collect connection server counter packets long
collect connection client counter bytes long
collect connection client counter packets long
collect application http host
!
!
flow monitor type performance-monitor reference-url_ipv4
record reference-url_ipv4
```

```
exporter reference-1
cache entries 5125
cache timeout synchronized 60 export-spread 15
history size 1
!
!
flow record type performance-monitor reference-url_ipv6
description ezPM record
match routing vrf input
match ipv6 protocol
match application name account-on-resolution
match connection client ipv6 address
match connection server transport port
match connection server ipv6 address
collect ipv6 dscp
collect ipv6 hop-limit
collect interface input
collect interface output
collect timestamp sys-uptime first
collect timestamp sys-uptime last
collect connection new-connections
collect connection sum-duration
collect application http uri statistics
collect connection delay response to-server sum
collect connection server counter responses
collect connection delay response to-server histogram late
collect connection delay network to-server sum
collect connection delay network to-client sum
collect connection client counter packets retransmitted
collect connection delay network client-to-server sum
collect connection delay application sum
collect connection delay application max
collect connection delay response client-to-server sum
collect connection transaction duration sum
collect connection transaction counter complete
collect connection server counter bytes long
collect connection server counter packets long
collect connection client counter bytes long
collect connection client counter packets long
collect application http host
!
!
flow monitor type performance-monitor reference-url_ipv6
record reference-url_ipv6
exporter reference-1
cache entries 5125
cache timeout synchronized 60 export-spread 15
history size 1
!
!Policy-maps
!============
policy-map type performance-monitor reference-in
parameter default account-on-resolution
class reference-app_ts
  flow monitor reference-app_ts_in
class reference-art_url_ipv4
  flow monitor reference-url_ipv4
class reference-art_url_ipv6
  flow monitor reference-url_ipv6
class reference-art_ipv4
  flow monitor reference-art_ipv4
class reference-art_ipv6
  flow monitor reference-art_ipv6
class reference-url_ipv4
  flow monitor reference-url_ipv4
class reference-url_ipv6
  flow monitor reference-url_ipv6
class reference-media_ipv4_in
  flow monitor reference-media_ipv4_in
class reference-media_ipv6_in
  flow monitor reference-media_ipv6_in
class reference-conv_ts_ipv4
  flow monitor reference-conv_ts_ipv4
```

```
class reference-conv_ts_ipv6
  flow monitor reference-conv_ts_ipv6
!
policy-map type performance-monitor reference-out
parameter default account-on-resolution
class reference-app_ts
  flow monitor reference-app_ts_out
class reference-art_url_ipv4
  flow monitor reference-url_ipv4
class reference-art_url_ipv6
  flow monitor reference-url_ipv6
class reference-art_ipv4
  flow monitor reference-art_ipv4
class reference-art_ipv6
  flow monitor reference-art_ipv6
class reference-url_ipv4
  flow monitor reference-url_ipv4
class reference-url_ipv6
  flow monitor reference-url_ipv6
class reference-media_ipv4_out
  flow monitor reference-media_ipv4_out
class reference-media_ipv6_out
  flow monitor reference-media_ipv6_out
class reference-conv_ts_ipv4
  flow monitor reference-conv_ts_ipv4
class reference-conv_ts_ipv6
  flow monitor reference-conv_ts_ipv6
!
!Interface Attachments
!=====================
interface GigabitEthernet0/0
service-policy type performance-monitor input reference-in
service-policy type performance-monitor output reference-out
!
```

Example: Configuring a Performance Monitor Context With Application Statistics Profile

The following example shows how to configure a performance monitor context with traffic monitor enabling per interface, application, client, and server statistics:

```
Device# configure terminal
Device(config)# performance monitor context perf-mon-test profile application-statistics
Device(config-perf-mon)# traffic-monitor application-client-server-stats cache-size 755
cache-type synchronized ipv6
Device(config-perf-mon)# exit
Device(config)# interface gigabitethernet 0/1
Device(config-if)# performance monitor context perf-mon-test
Device(config-if)# exit
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases

Related Topic	Document Title
Cisco Application Visibility and Control User Guide	Cisco Application Visibility and Control User Guide for Cisco IOS XE Release 3.11S and Cisco IOS Release15.4(1)T. Cisco Application Visibility and Control User Guide for Cisco IOS XE Release 3.12S and Cisco IOS Release15.4(2)T. Cisco Application Visibility and Control User Guide for Cisco IOS XE Release 3.13S and Cisco IOS Release15.4(3)T.

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Easy Performance Monitor

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to . An account on Cisco.com is not required.

Table 8: Feature Information for Easy Performance Monitor

Feature Name	Releases	Feature Information
Easy Performance Monitor	15.4(1)T	The Easy Performance Monitor chapter describes how to configure Easy Performance Monitor (ezPM) for Application Visibility and Control (AVC). This feature was introduced.

Feature Name	Releases	Feature Information
Adaptive AVC Reporting	15.4(3)T	The support for Application Statistics profile was added. The following command was modified: **performance monitor** The **application-statistics** keyword was added.