



Media Server 2000 Series Performance Management

Performance Measurements are available through an SNMP interface and can be polled at scheduled intervals by an external poller or utility in the management server or other off board system. The media server provides two types of performance measurements, Gauges and Counters.

Gauges represent the current state of activities on the media server. Gauges can decrease and increase in value. The value of a gauge is a snapshot of the current activity on the media server at that moment.

Counters always increase in value and are cumulative. Counters never decrease in value unless the server is reset. Then the counters begin again at zero.

Note: Media Server 2000 performance measurements are collected and displayed by the IEMS system beginning in SN07.

Performance data

The media server performance measurements are provided by two MIBs (acPerfMediaGateway and acPerfMediaServices). The first mib is generic-type of PM mib. The second PM mib is media server specific.

The generic PM mib covers the:

- control protocol,
- RTP stream, and
- system packets statistics.

The media specific PM mib covers the:

- interactive voice response (IVR),
- bearer channel tandeming (BCT),
- conferencing (CONF), and
- test trunks (TT) statistics.

Control Protocol performance data

The following table contains a brief description of the MS2000 series Control Protocol Performance Measurements.

Name	Type	Description
acPerfCpNumDupsFor CompletedTransactions	Counter	Number of Duplicated Completed Transactions: The number of times a duplicate transaction request was received after the initial transaction had already been completed. The gateway resends the response for this transaction.
acPerfCpNumDupsFor OutstandingTransactions	Counter	Number of Duplicated Transactions Outstanding. The number of times a duplicate transaction request was received while the initial transaction was outstanding (still in progress). The gateway ignores the duplicate request.
acPerfCpMessageSend Successes	Counter	Messages Send Successes. Number of times there was a success in sending a call control (H.248) message. Call control messages are sent using the system's socket library. This counter tracks successes in using the local socket services. It does not track successes in end-to-end message transfer between the gateway and the call agent.
acPerfCpMessageSend Errors	Counter	Message Send Errors. Number of times there was a failure in sending a call control (H.248) message. The message is sent by way of a datagram using the system's socket library. Normally a failure on a socket send operation would be attributed to an internal system problem.

Name	Type	Description
acPerfCpMessageReceive Successes	Counter	Message Receive Successes. Number of times there was a success in receiving a call control (H.248) message. Call control messages are received using the system's socket library. This counter tracks successes in using the local socket services. It does not track successes in end-to-end message transfer between the gateway and the call agent.
acPerfCpMessageReceive Errors	Counter	Message Receive Errors. Number of times there was a failure in receiving a call control(H.248) message. Call control messages are received using the system's socket library. A failure on the socket receive operation can be attributed to an internal system problem or with the call agent sending a message larger than what is supported by the gateway.
acPerfCpProtocolSyntax Errors	Counter	Protocol Syntax Errors. Number of syntax errors detected in incoming call control (H.248) messages.
acPerfCpMessage Retransmissions	Counter	Message Retransmissions. Each time the call engine times out waiting for an acknowledgement it retransmits the control protocol message, unless the number of max retransmissions is exceeded. This counter is incremented each time a message is retransmitted due to a timeout.

Name	Type	Description
acPerfCpMessageMax RetransmissionsExceeded	Counter	Message Max Re-transmissions Exceeded. Number of times the call control message maximum retransmission count was exceeded. The gateway attempted several times to send a message to the call agent, but each time, an ack was not received. A failure of this type results in a failed call and is usually an indication that subsequent calls will fail. This problem is typically a result of the call agent being down or the result of a network problem.
acPerfCpMessagesFrom UntrustedSources	Counter	Messages From Untrusted Sources. Number of messages received from untrusted sources, that is from network nodes other than the node on which the call agent is running.

RTP stream performance data

The following table contains a brief description of the MS2000 series RTP stream Performance Measurements.

Name	Type	Description
acPerfRtpSenderPackets	Counter	RTP Sender Packets (Card). Total number of RTP packets sent by the system for this card.
acPerfRtpSenderOctets	Counter	RTP Sending Octets (Card). Total number of non-header RTP octets sent by this card.
acPerfRtpReceiverPackets	Counter	RTP Receiver Packets (Card). Total number of RTP packets received by the system for this card.
acPerfRtpReceiverOctets	Counter	RTP Receiver Octets (Card). Total number of non-header RTP octets received by this card.
acPerfRtpRcvrLostPackets	Counter	RTP Receiver Lost Packets (Card). Total number of RTP packets lost as observed by this card.

Name	Type	Description
acPerfRtpFailedDueToLackOfResources	Counter	RTP Failed Due to Lack of Resources. The number of times an rtp request was rejected due to lack of resources since the last application restart.
acPerfRtpSimplexInSessionsTotal	Counter	RTP Simplex In Session Total. Total number of simplex input RTP sessions. A simplex (one-way) session would be used to play an announcement.
acPerfRtpSimplexInSessionsCurrent	Gauge	RTP Simplex In Sessions. Current number of simplex input RTP sessions.
acPerfRtpSimplexOutSessionsTotal	Counter	RTP Simplex Out Session Total. Total number of simplex output RTP sessions.
acPerfRtpSimplexOutSessionsCurrent	Gauge	RTP Simplex Out Session Current. Current number of simplex output RTP sessions.
acPerfRtpDuplexSessionsTotal	Counter	RTP Duplex Sessions Total. Total number of duplex RTP sessions.
acPerfRtpDuplexSessionsCurrent	Gauge	RTP Duplex Sessions Current. Current number of duplex RTP sessions.

System packets performance data

The following table contains a brief description of the MS2000 series system packets Performance Measurements.

Name	Type	Description
acPerfSystemPacketEndpoints	Gauge	System Packet Endpoints. Number of endpoints reserved for all packet network-related functions (conferencing, plays, etc.).
acPerfSystemPacketEndpointsInUse	Gauge	System Packets Endpoints In Use. Number of endpoints that the call engine is currently using for all packet network-related functions (conferencing, plays, etc.).

IVR performance data

The following table contains a brief description of the MS2000 series interactive voice response Performance Measurements.

Name	Type	Description
acPerflvrPlayRequests	Counter	Play Requests: The total number of announcement requests received from the call agent.
acPerflvrPlaySuccessful	Counter	Play Successes: Number of announcement requests processed successfully.
acPerflvrPlayFailedDueToLackOfResources	Counter	Play Failures Due to Lack of Resources: Number of announcement requests which failed to be played because some resource was not available.
acPerflvrPlayInProgress	Gauge	Play In Progress: The number of announcement operations that are currently in progress.
acPerflvrPlayDuration	Counter	Play Duration: The duration, in seconds, of all successful announcement requests. The average duration of all requests (average hold time) can be computed by dividing acPerfPlayDuration by acPerfPlaySuccessful.
acPerflvrPlayFailedDueToProvMismatch	Counter	Play Failed Due to Provisioning Mismatch: Number of audio segments which failed to be played because of a provisioning mismatch.
acPerflvrPlayCollectRequests	Counter	Play Collect Requests: The number of play collect requests.
acPerflvrPlayCollectSuccessful	Counter	Play Collect Successful: Number of play collects completed successfully. A request is considered successful if the entire sequence, from initial prompt to success/failure prompt, is played out without a failure due to lack of resources, provisioning mismatch, or any other media server failure. The failure of the user to enter the proper digits is not a reason for failure of the request.

Name	Type	Description
acPerflvrPlayCollectFailedDueToLackOfResources	Counter	Play Collect Failure due to Lack of Resources: The number of play collect requests that failed due to lack of resources.
acPerflvrPlayCollectFailedDueToProvMismatch	Counter	Play Collect Failed Due to Provisioning Mismatches: The number of play collect requests that failed due a provisioning mismatch.
acPerflvrPlayCollectInProgress	Gauge	Play Collect In Progress: The number of play collect operations that are currently in progress.
acPerflvrPlayCollectDuration	Counter	Play Collect Duration in Seconds: The duration, in seconds, of all successful play collect requests. The average duration of all requests can be computed by dividing acPlayCollectDuration by acPlayCollectSuccessful.
acPerflvrContDigitCollectRequests	Counter	Number of Continuous Digit Collection Requests: The number of continuous digit collect requests.
acPerflvrContDigitCollectSuccessful	Counter	Number of Continuous Digit Collect Requests Successful: Number of continuous digit collects completed successfully.
acPerflvrContDigitCollectFailedDueToLackOfResources	Counter	Number of Continuous Digit Collection request failed (no resources): The number of continuous digit collect requests that failed due to lack of resources.
acPerflvrContDigitCollectInProgress	Gauge	In Progress Continuous Digit Collection requests: The number of continuous digit collect operations that are currently in progress.
acPerflvrContDigitCollectDuration	Counter	Successful Continuous Digit Collect Requests: The duration, in seconds, of all successful continuous digit collect requests. The average duration of all requests can be computed by dividing acDCCollectDuration by acDCCollectSuccessful.

BCT performance data

The following table contains a brief description of the MS2000 series bearer channel tandeming Performance Measurements.

Name	Type	Description
acPerfBctRequests	Counter	Total BCT Requests: The total number of BCT contexts opened since the media server initialized.
acPerfBctSuccessful	Counter	Successful BCT requests: Number of BCT requests processed successfully.
acPerfBctFailedDueToLackOfResources	Counter	Failed BCT requests: Number of BCT requests which failed to be played because a resource was not available.
acPerfBctInProgress	Gauge	In progress BCT requests: The number of BCT calls that are currently in progress.
acPerfBctDuration	Counter	Duration of BCT contexts: The duration, in seconds, of all successful BCT contexts. The average duration of all contexts can be computed by dividing acPerfBCTDuration by acPerfBCTSuccessful.
cPerfBctTotalParticipants	Counter	Total of BCT participants: The total number of BCT participants since the media server initialized.
acPerfBctCurrentNumberOfParticipants	Gauge	In progress BCT participants: The number of participants in all BCT calls that are currently in progress.

CONF performance data

The following table contains a brief description of the MS2000 series conferencing Performance Measurements.

Name	Type	Description
acPerfConfRequests	Counter	Total number of conferences processed: The total number of conferences processed since the last application restart. This is the number of conferences created and not the number of members.
acPerfConfSuccessful	Counter	Successful Conferences: Number of conference requests processed successfully.
acPerfConfInProgress	Gauge	In-progress Conferences: The number of conferences currently in progress.
acPerfConfDuration	Counter	Conference Duration: The duration, in seconds, of all successful conference requests. The average duration of all requests can be computed by dividing acPerfConfDuration by acPerfConfSuccessful.
acPerfConfFailedDueToLackOfResources	Counter	Conference Failures due to lack of resources: The number of times a conference request was rejected due to lack of resources since the last application restart.
acPerfConfAddRequests	Counter	Requests to add a conferee: The number of requests to add a conferee to an existing conference.
acPerfConfAddSuccessful	Counter	Conferee adds successful: The number of times a conferee was added successfully to an existing conference.
acPerfConfAddFailedDueToLackOfResources	Counter	Conferee Add Failures due to lack of resources: The number of times a conferee could not be added to an existing conference due to a lack of resources.

Name	Type	Description
acPerfConfPortsUsed	Counter	Conference Ports Reserved: The total number of ports that were reserved for conferences since the last application restart.
acPerfConfPortsReserved	Gauge	Conference Port Reserved for monitor: A monitor port may be reserved by the call agent when setting up a conference. The monitor port can then be used by one or more listen-only conferees.

TT performance data

The following table contains a brief description of the MS2000 series test trunks Performance Measurements.

Name	Type	Description
acPerfTtRequests	Counter	Test Trunk Test Requests: The number of test trunk tests requested.
acPerfTtSuccessful	Counter	Test Trunk Test Request Successes: The number of test trunk tests that were successfully setup and torn down. This PM is not a reflection on whether the test actually passed or failed.
acPerfTtInProgress	Gauge	In-Progress Test Trunk Tests: The number of test trunk tests that are currently in progress.
acPerfTtDuration	Counter	Test Trunk Test Duration: The duration, in seconds, of all successful test trunk requests. This number is the time between a test trunk test being initiated and completed. The average duration of all requests can be computed by dividing acPerfTtDuration by acPerfTtSuccessful.
acPerfTtFailedDueToLackOfResources	Counter	Test Trunk failures due to lack of resources: The number of times a test trunk request was rejected due to lack of resources since the last application restart.