# Performance Counters

# Import Unified CCE Data Collector Set Template

This chapter describes performance counters supported for Microsoft Windows Performance Monitor to monitor Unified CCE components.

In the Performance Monitor, use the template file, **CCE.xml**, to create a Data Collector Set to capture the standard set of performance counters for the Unified CCE components. On any machine where a Unified CCE component is installed, the template file is installed in the `icm\serviceability\perfmon` directory. After you create a Data Collector set, you can schedule or manually start it to capture the Unified CCE performance counters described in this chapter.

The performance counter log files that this Data Collector Set generates are created as CSV files.

**Note**    The template file may contain component-specific counters for Unified CCE components that are not installed on your machine. Counters for these components are included in the log files with blank values.

Follow these steps to import the template file and create the Data Collector set.

**Procedure**

**Step 1**    Start the 32-bit Windows Performance Monitor tool by using the shortcut called Performance Monitor. This shortcut is available in the Cisco Unified CCE Tools folder. Alternatively, you can also launch the 32-bit utility by running the command *mmc /32 perfmon.msc*.

**Step 2**    In the panel on the left, expand **Data Collector Sets**.

**Step 3**    Right click **User Defined** and select **New** > **Data Collector Set**.
The Data Collector Set wizard opens.

**Step 4**    Provide a name for the Data Collector Set and select **Create from a template**. Click **Next**.

**Step 5**    Click **Browse**. Go to  `icm\serviceability\perfmon` and select the file **CCE.xml**. Click **Next**.

Step 6    Specify the location where you want to collect the performance counter log. Click **Next**.

Step 7    Select **Save and Close** and click **Finish**.

The Performance Monitor creates the Data Collector Set by importing the XML file. You can edit the Data Collector Set to modify the component monitoring, for example to change the sampling interval or add or remove counters. For more information, see "Creating Data Collector Sets" in the Performance Monitor online help.

# Platform Health Monitoring Counters

The following table lists the performance counters that you should watch on a regular basis to determine the health of the contact center application.

Threshold values are not monitored by the application itself – alarms are not generated if threshold are exceeded. The responsibility for polling and threshold alarming is extended to the management station.

*Table 1: Performance Counters - Health Monitoring*

| Counter Name (Instance) | Property | Value |
|---|---|---|
| % Processor Time (_Total) | Performance Object | Processor |
| | Type | Int32 |
| | Units (Range) | Percentage (0 - 100%) |
| | Threshold (Green) | < 50% |
| | Threshold (Yellow) | 50% - 60% |
| | Threshold (Red) | > 60% (sustained) |
| | Description | Primary indicator of processor activity; displays the average percentage of CPU busy time observed during the sample interval. |
| Processor Queue Length | Performance Object | System |
| | Type | Int32 |
| | Units (Range) | # threads |
| | Threshold (Green) | < 2 * #CPUs |
| | Threshold (Yellow) | — |
| | Threshold (Red) | >= 2 * #CPUs (sustained) |
| | Description | Number of threads in the processor queue waiting to be serviced. Microsoft states that Processor Queue Length is OK up to 10 per CPU. This may be the case for non-real time applications but Unified CC performance is impacted if this queue length is excessive for a sustained period of time. Timeouts are likely if the server becomes CPU bound or a single application (or process) monopolizes the CPU. |

| Counter Name (Instance) | Property | Value |
|---|---|---|
| Available Bytes | Performance Object | Memory |
| | Type | Int32 |
| | Units (Range) | Percentage (0 - 100%) |
| | Threshold (Green) | > 30% |
| | Threshold (Yellow) | 20% - 30% |
| | Threshold (Red) | < 20% |
| | Description | Amount of physical memory available to running processes; threshold values are a percentage of physical memory. This is a snap shot–not a running average. Sustained samples below 20% (available) may be indicative of a memory leak. |
| Pages / sec | Performance Object | Memory |
| | Type | Int32 |
| | Units (Range) | # page faults |
| | Threshold (Green) | < 10 |
| | Threshold (Yellow) | >= 10 |
| | Threshold (Red) | > 10 (sustained) |
| | Description | Pages/sec is the rate at which pages are read from or written to disk to resolve hard page faults. Excessive page faults adversely impacts performance – root cause must be investigated. |
| Avg. Disk Queue Length (_Total) | Performance Object | Physical Disk |
| | Type | Float |
| | Units (Range) | Average # read/write requests |
| | Threshold (Green) | < 1.5 |
| | Threshold (Yellow) | — |
| | Threshold (Red) | >= 1.5 (sustained) |
| | Description | Average number of both read and write requests that were queued for the selected disk during the sample interval. |
| % Disk Time (_Total) | Performance Object | Physical Disk |
| | Type | Int32 |
| | Units (Range) | Percentage (0 - 100%) |
| | Threshold (Green) | < 60% |
| | Threshold (Yellow) | 60% - 80% |
| | Threshold (Red) | > 80% |
| | Description | Percentage of elapsed time that the disk drive was busy servicing read or write requests. |

| Counter Name (Instance) | Property | Value |
|---|---|---|
| Bytes Total/sec | Performance Object | Network Interface |
| | Type | Int32 |
| | Units (Range) | Percentage (0 - 100%) |
| | Threshold (Green) | < 25% |
| | Threshold (Yellow) | 25% - 30% |
| | Threshold (Red) | > 30% |
| | Description | Rate at which bytes are sent and received over each network adapter. Threshold values are a percentage of available bandwidth. |
| Output Queue Length | Performance Object | Network Interface |
| | Type | Int32 |
| | Units (Range) | # packets in queue |
| | Threshold (Green) | 0 |
| | Threshold (Yellow) | 1 |
| | Threshold (Red) | > 1 (sustained) |
| | Description | Length of the output packet queue (in packets). If too large, there are delays and the bottleneck should be found and eliminated. |
| Buffer cache hit ratio | Performance Object | SQLServer:Buffer Manager |
| | Type | Int32 |
| | Units (Range) | Percentage (0 - 100%) |
| | Threshold (Green) | > 90% |
| | Threshold (Yellow) | — |
| | Threshold (Red) | < 90% |
| | Description | This counter shows the percentage of pages in the buffer pool without needing to read from disk. Thresholds are expressed as a percentage of hits; instances in which the requested page was found in the cache. |
| | | This counter is typically a good indicator of whether there is sufficient RAM installed in the server. |
| | | If you are using SQL Server Standard Edition in a large enterprise or hosted environment and this counter (as well as other performance counters) is not within the correct range, upgrading SQL Server to Enterprise Edition may be the next step. Upgrading SQL Server to Enterprise Edition requires an upgrade of the operating system to Windows Server Enterprise Edition. See Microsoft documentation for details. |

# Platform Diagnostic Counters – Automatic Collection

The Node Manager samples and collects counter values automatically. The counter values are stored in a disk file on the server and are sampled at one-minute intervals.

Data files contain a rolling window of counter values—older data is discarded in place of new data. Data is stored in multiple files (with a maximum size of 1 MB each) and saved for a maximum of 45 days.

### Platform Diagnostic Counter Values

**Data file location**
```
\icm\logs
```
**File naming convention**
```
Perf_MACHINENAME_YYYYMMDDHHMMSS.CSV
```
**Where**

- *MACHINENAME* is the assigned Windows computer name.
- *YYYYMMDD* is the year, month, day the file was created.
- *HHMMSS* is the hour:minute:second the file was created.

Analysis of these counter values is beneficial when diagnosing a problem with a Unified CCE application component.

*Table 2: Performance Counters - Diagnostics*

| Counter Name | Property | Value |
|---|---|---|
| % Processor Time (_Total) | Component | Processor |
| | Type | Int32 |
| | Units (Range) | Percentage (0 – 100%) |
| | Description | % Processor Time is the percentage of elapsed time that the processor spends to run a non-Idle thread. It is calculated by measuring the duration that the idle thread is active in the sample interval, and subtracting that time from interval duration. (Each processor has an idle thread that consumes cycles when no other threads are ready to run.) This counter is the primary indicator of processor activity and displays the average percentage of busy time observed during the sample interval. It is calculated by monitoring the time that the service is inactive and subtracting that value from 100%. |
| Handle Count (_Total) | Component | Process |
| | Type | Int32 |
| | Units (Range) | # handles |
| | Description | The total count of handles currently open by this process. This number is equal to the sum of the handles currently open by each thread in this process. |

| Counter Name | Property | Value |
|---|---|---|
| Page Faults / sec | Component | Memory |
| | Type | Int32 |
| | Units (Range) | # faults |
| | Description | Page Faults/sec is the average number of pages faulted per second. It is measured in number of pages faulted per second because only one page is faulted in each fault operation; hence, this is also equal to the number of page fault operations. This counter includes both hard faults (those that require disk access) and soft faults (where the faulted page is found elsewhere in physical memory). Most processors can handle large numbers of soft faults without significant consequence. However, hard faults, which require disk access, can cause significant delays. |
| Committed Bytes | Component | Memory |
| | Type | Int32 |
| | Units (Range) | # bytes |
| | Description | Committed Bytes is the amount of committed virtual memory, in bytes. Committed memory is the physical memory that has space reserved on the disk paging files. There can be one or more paging files on each physical drive. This counter displays the last observed value only; it is not an average. |
| Pages / sec | Component | Memory |
| | Type | float |
| | Units (Range) | # pages per second |
| | Description | Pages/sec is the number of pages either read from the disk or written to the disk to resolve memory references to pages that were not in memory at the time of the reference. Pages/sec is the sum of Pages Input/sec and Pages Output/sec. This counter includes paging traffic on behalf of the system cache to access file data for applications. This is also the primary counter to observe if you are concerned about excessive memory pressure (thrashing) and the excessive paging that may result. This counter, however, also accounts for such activity as the sequential reading of memory mapped files, whether cached or not. |
| Threads | Component | System |
| | Type | Int32 |
| | Units (Range) | # threads |
| | Description | Threads is the number of threads in the computer at the time of data collection. This is an instantaneous count, not an average over the time interval. A thread is the basic executable entity that can run instructions in a processor. |

| Counter Name | Property | Value |
|---|---|---|
| Processor Queue Length | Component | System |
| | Type | Int32 |
| | Units (Range) | # threads |
| | Description | Processor Queue Length is the number of threads in the processor queue. Unlike the disk counters, this counter shows ready threads only, not threads that are running. There is a single queue for processor time even on computers with multiple processors. Therefore, if a computer has multiple processors, you must divide this value by the number of processors servicing the workload. A sustained processor queue of fewer than 10 threads per processor is generally acceptable, dependent on the workload. |
| Processes | Component | System |
| | Type | Int32 |
| | Units (Range) | # processes |
| | Description | Processes is the number of processes in the computer at the time of data collection. This is an instantaneous count, not an average over the time interval. Each process represents the running of a program. |

*Table 3: Performance Counters - Diagnostics*

| Counter Name | Property | Value |
|---|---|---|
| % Processor Time (_Total) | Component | Processor |
| | Type | Int32 |
| | Units (Range) | Percentage (0 – 100%) |
| | Description | % Processor Time is the percentage of elapsed time that the processor spends to run a non-Idle thread. It is calculated by measuring the duration that the idle thread is active in the sample interval, and subtracting that time from interval duration. (Each processor has an idle thread that consumes cycles when no other threads are ready to run.) This counter is the primary indicator of processor activity and displays the average percentage of busy time observed during the sample interval. It is calculated by monitoring the time that the service is inactive and subtracting that value from 100%. |
| Handle Count (_Total) | Component | Process |
| | Type | Int32 |
| | Units (Range) | # handles |
| | Description | The total count of handles currently open by this process. This number is equal to the sum of the handles currently open by each thread in this process. |

| Counter Name | Property | Value |
|---|---|---|
| Page Faults / sec | Component | Memory |
| | Type | Int32 |
| | Units (Range) | # faults |
| | Description | Page Faults/sec is the average number of pages faulted per second. It is measured in number of pages faulted per second because only one page is faulted in each fault operation; hence, this is also equal to the number of page fault operations. This counter includes both hard faults (those that require disk access) and soft faults (where the faulted page is found elsewhere in physical memory). Most processors can handle large numbers of soft faults without significant consequence. However, hard faults, which require disk access, can cause significant delays. |
| Committed Bytes | Component | Memory |
| | Type | Int32 |
| | Units (Range) | # bytes |
| | Description | Committed Bytes is the amount of committed virtual memory, in bytes. Committed memory is the physical memory that has space reserved on the disk paging files. There can be one or more paging files on each physical drive. This counter displays the last observed value only; it is not an average. |
| Pages / sec | Component | Memory |
| | Type | float |
| | Units (Range) | # pages per second |
| | Description | Pages/sec is the number of pages either read from the disk or written to the disk to resolve memory references to pages that were not in memory at the time of the reference. Pages/sec is the sum of Pages Input/sec and Pages Output/sec. This counter includes paging traffic on behalf of the system cache to access file data for applications. This is also the primary counter to observe if you are concerned about excessive memory pressure (thrashing) and the excessive paging that may result. This counter, however, also accounts for such activity as the sequential reading of memory mapped files, whether cached or not. |
| Threads | Component | System |
| | Type | Int32 |
| | Units (Range) | # threads |
| | Description | Threads is the number of threads in the computer at the time of data collection. This is an instantaneous count, not an average over the time interval. A thread is the basic executable entity that can run instructions in a processor. |

| Counter Name | Property | Value |
|---|---|---|
| Processor Queue Length | Component | System |
| | Type | Int32 |
| | Units (Range) | # threads |
| | Description | Processor Queue Length is the number of threads in the processor queue. Unlike the disk counters, this counter shows ready threads only, not threads that are running. There is a single queue for processor time even on computers with multiple processors. Therefore, if a computer has multiple processors, you must divide this value by the number of processors servicing the workload. A sustained processor queue of fewer than 10 threads per processor is generally acceptable, dependent on the workload. |
| Processes | Component | System |
| | Type | Int32 |
| | Units (Range) | # processes |
| | Description | Processes is the number of processes in the computer at the time of data collection. This is an instantaneous count, not an average over the time interval. Each process represents the running of a program. |
| % Idle Time (0 C:) | Component | Physical Disk |
| | Type | Float |
| | Units (Range) | Percentage (0 - 100%) |
| | Description | % Idle Time reports the percentage of time during the sample interval that the disk was idle. **Note** This instance is for the virtual machine's drive C—the drive where the software is installed and where logs are stored. |
| % Idle Time (1 E:) | Component | Physical Disk |
| | Type | Float |
| | Units (Range) | Percentage (0 - 100%) |
| | Description | % Idle Time reports the percentage of time during the sample interval that the disk was idle. **Note** This instance is for the virtual machine's drive E—the drive where the database is stored. If the component on this virtual machine does not have a database component, the counter sample values appear as Error, but this has no adverse effect on the component itself. |

| Counter Name | Property | Value |
|---|---|---|
| Avg. Disk Queue Length (0 C:) | Component | Physical Disk |
| | Type | Float |
| | Units | # requests |
| | Description | Avg. Disk Queue Length is the average number of both read and write requests that were queued for the selected disk during the sample interval.<br><br>**Note**      This instance is for the virtual machine's drive C—the drive where the software is installed and where logs are stored. |
| Avg. Disk Queue Length (1 E:) | Component | Physical Disk |
| | Type | Float |
| | Units (Range) | # requests |
| | Description | Avg. Disk Queue Length is the average number of both read and write requests that were queued for the selected disk during the sample interval.<br><br>**Note**      This instance is for the virtual machine's drive E—the drive where the database is stored. If the component on this virtual machine does not have a database component, the counter sample values appear as `Error`, but this has no adverse effect on the component itself. |
| Avg. Disk sec/Read (0 C:) | Component | Physical Disk |
| | Type | Float |
| | Units | seconds |
| | Description | Avg. Disk sec/Read is the average time, in seconds, of a read of data from the disk.<br><br>**Note**      This instance is for the virtual machine's drive C—the drive where the software is installed and where logs are stored. |
| Avg. Disk sec/Read (1 E:) | Component | Physical Disk |
| | Type | Float |
| | Units | seconds |
| | Description | Avg. Disk sec/Read is the average time, in seconds, of a read of data from the disk.<br><br>**Note**      This instance is for the virtual machine's drive E—the drive where the database is stored. If the component on this virtual machine does not have a database component, the counter sample values appear as `Error`, but this has no adverse effect on the component itself. |

| Counter Name | Property | Value |
|---|---|---|
| Avg. Disk sec/Write (0 C:) | Component | Physical Disk |
| | Type | Float |
| | Units | seconds |
| | Description | Avg. Disk sec/Write is the average time, in seconds, of a write of data to the disk.<br><br>**Note** This instance is for the virtual machine's drive C—the drive where the software is installed and where logs are stored. |
| Avg. Disk sec/Write (1 E:) | Component | Physical Disk |
| | Type | Float |
| | Units | seconds |
| | Description | Avg. Disk sec/Write is the average time, in seconds, of a write of data to the disk.<br><br>**Note** This instance is for the virtual machine's drive E—the drive where the database is stored. If the component on this virtual machine does not have a database component, the counter sample values appear as `Error`, but this has no adverse effect on the component itself. |

# Platform Diagnostic Counters

## All Components

If a problem occurs on a Unified CCE/Unified ICM component, to further diagnose the problem, enable these counters using the Windows PerfMon tool (On Windows server, Start > Cisco Unified CCE Tools > Performance Monitor). At first, set the interval to 15 seconds and collect a sample large enough before, during, and after the problem. Save the data in .CSV format for simple import into Microsoft Office Excel. Attach the file to the TAC case.

If the data does not provide enough resolution to diagnose root cause, increase the interval to 5 seconds. A sample interval more frequent than 3 seconds should not be attempted.

*Table 4: Diagnostic Counters - All Components*

| Performance Object | Instance | Counter Name |
|---|---|---|
| LogicalDisk | _Total | Avg. Disk Queue Length |
| LogicalDisk | C: | Avg. Disk Queue Length |
| LogicalDisk | <> | Avg. Disk Queue Length |
| Network Interface | <NIC Name> | Packets Outbound Discarded |
| PhysicalDisk | _Total | Disk Transfers/sec |
| Process | _Total | Page Faults/sec |

| Performance Object | Instance | Counter Name |
|---|---|---|
| Process | _Total | Virtual Bytes |
| Process | _Total | Working Set |
| Processor | _Total | Interrupts/sec |
| Process | <virus scanner> | % Processor Time |
| Process | <virus scanner> | Page Faults/sec |
| Process | <virus scanner> | Virtual Bytes |
| Process | <virus scanner> | Working Set |

## Logger/Administration & Data Server/HDS

These counters are intended for Unified CCE/Unified ICM components that have a SQL Server database installed. SQL Server counters are listed in the next session.

Set the initial sample frequency to 15 seconds. If the resolution is insufficient, decrease the frequency to 5 seconds.

*Table 5: Diagnostic Counters - Logger, Administration & Data Server, and HDS*

| Performance Object | Instance | Counter Name |
|---|---|---|
| Physical Disk | <> | % Disk Time |
| Physical Disk | <> | Avg. Disk Queue Length |
| Physical Disk | <> | Disk Transfers/sec |
| Process | ** See note | % Processor Time |
| Process | ** See note | Page Faults/sec |
| Process | ** See note | Virtual Bytes |
| Process | ** See note | Working Set |
| Process | sqlservr | % Processor Time |
| Process | sqlservr | Page Faults/sec |
| Process | sqlservr | Virtual Bytes |
| Process | sqlservr | Working Set |

**Note** Logger Processes: configlogger, histlogger, recovery, replication

AW/HDS Processes: configlogger, replication, rtclient, rtdist

## SQL Server

The listed counters are available on those servers on which a Unified CCE/Unified ICM database is installed.

Set the initial sample frequency to 15 seconds. If the resolution is insufficient, decrease the frequency to 5 seconds.

*Table 6: Diagnostic Counters - SQL Server*

| Performance Object | Instance | Counter Name |
|---|---|---|
| SQLServer:Access Methods | | Full Scans/sec |
| SQLServer:Buffer Manager | | Buffer cache hit ratio |
| SQLServer:Buffer Manager | | Page reads/sec |
| SQLServer:Buffer Manager | | Page writes/sec |
| SQLServer:Buffer Manager | | Stolen pages |
| SQLServer:Databases | _Total | Transactions/sec |
| SQLServer:Databases | csco_awdb[1] | Transactions/sec |
| SQLServer:Databases | csco_hds[2] | Transactions/sec |
| SQLServer:General Statistics | | User Connections |
| SQLServer:Latches | | Average Latch Wait Time (ms) |
| SQLServer:Locks | _Total | Lock Timeouts/sec |
| SQLServer:Locks | _Total | Number of Deadlocks/sec |
| SQLServer:Memory Manager | | Memory Grants Pending |

[1] Where "csco" is the Unified ICM/Unified CCE instance name.
[2] Where "csco" is the Unified ICM/Unified CCE instance name.

# Component-Specific Counters

Performance counters that measure time durations in milliseconds, provide granular measurements of at least 16 milliseconds.

**Note**  To enable a counter that is disabled by default, make a change to the registry.

Performance counter-objects that are being captured or monitored as "per second" or rate values, are interpreted as average number of operations completed during each second of the sample interval. This is a computed value, and the performance monitor tool essentially uses the following formula to represent the counter value.

All such **perfmon counters** that represent rate values are defined as either "PERF_COUNTER_COUNTER" or "PERF_COUNTER_BULK_COUNT" type.

### PERF_COUNTER_COUNTER / PERF_COUNTER_BULK_COUNT Calculations:

*Table 7: Calculating PERF_COUNTER_COUNTER or PERF_COUNTER_BULK_COUNT*

| PERF_COUNTER_COUNTER= (N1- N0) / ((D1 - D0) / F) | |
|---|---|
| Numerator (N) | The numerator (N) represents the number of operations performed during the last sample interval |
| Denominator (D) | The denominator (D) represents the number of ticks elapsed during the last sample interval |
| F | F is the frequency of the ticks. |

**Note**  For example, if the VRU PIM **perfmon counter** for "New Calls/sec" is set up to capture the counter value in every 5 seconds, and during that interval the VRU PIM receives 15 calls, then the rate value shown is 3 calls / sec.

# Router

**Performance Object**
   Cisco ICM Router

### Counter Instance

"{ICM Instance Name}" – if multiple instances installed

*Table 8: Router Performance Counters*

| Always ON? | Counter Name | Description |
|---|---|---|
| Y | Agents Logged On<br><br>These counters are also quite useful for long-term trending to determine whether there are capacity issues now or whether there are in the future. The counter values can be compared to other PerfMon counters (for example, CPU, Memory, Disk, and NIC). Relationships and cause/effect analysis can greatly assist in confirming existing or predicting upcoming capacity/performance problems. | The number of (contact center) agents currently logged in. |
| Y | Calls In Progress | The number of calls currently in progress (being controlled by the CCE application). |
| Y | Active Agent Answers Enabled Calls | Total number of active calls on which Agent Answers is enabled, for a CCE instance. The Agent Answers service silently monitors a call and displays relevant real-time suggestions to agents and supervisors in Finesse gadgets. These suggestions enrich an agent's ability to respond to a caller. |
| Y | Calls/sec | The (calculated) inbound call rate measured in the number of calls received per second. |
| Y | Calls In Queue | The number of calls queued in all network Voice Response Units (VRUs), from the router's perspective, including those calls that are in the process of transferring to the VRU for queuing. |
| Y | Calls In Router | The number of active calls in the router, including the calls sent to VRU for treatment or queuing and the calls the router is waiting for response from the routing client. |
| Y | CSNRequestsPerSec | The number of contact share node requests received per second. |
| Y | CSNRequestsAvgRespTime | Average time taken to respond to contact share node requests. Time is mentioned in milliseconds (ms). |
| Y | CSNNumberSGUpdates | Total Number of Skill Group updates from the target CCE systems at every Live Data-published interval (default 3 sec). |
| Y | CSNNumberPQUpdates | Total Number of PQ updates from the target CCE systems at every Live Data-published interval (default 3 sec). |
| Y | Pending PQ Count | Total number of precision queue configuration operations in the system yet to be fully processed. |
| Y | Pending PQ Agent Count | Total number of agents yet to be evaluated or handled following one or more precision queue configuration operations. |

| Always ON? | Counter Name | Description |
|---|---|---|
| Y | Average PQ Update Time | Average time to completely process a precision queue configuration update. Average time is calculated as the moving average for the last 10 precision queue configuration updates. |
| N | Size | The current Router state size - thetotal size ofall of the state transfer objects in Router memory; this size is measured in kilobytes.After one Router side goes out of service, when it returns in-service,the Router state is transferred from the surviving Router side to the returning Router side. |
| N | Messages Processed/sec | The number of MDS messages that the Router processed. By default, this counter is disabled. |
| N | Bytes Processed/sec | The rate at which the Router processed the data bytes. By default, this counter is disabled. |
| N | Avg Process Time/Message (ms) | The average time (in milliseconds) that the Router spends processing an MDS message. |
| N | Max Process Time(ms) | The maximum time (in milliseconds) that the Router spends processing an MDS message. |

### Enable Optional Counters

**Key**
> HKEY_LOCAL_MACHINE SOFTWARE\Cisco Systems,
> Inc.\ICM\*<Instance>*\*<node>*\Router\CurrentVersion\Debug

**Name**
> PerfmonCounterInterval

**Type**
> REG_DWORD

**Default**
> 0

**Enabled**
> 1

# Logger

**Performance Object**
> Cisco ICM Logger

**Counter Instance**
> "{ICM Instance Name}" – if multiple instances installed

*Table 9: Logger Performance Counters*

| Always ON? | Counter Name | Description |
|---|---|---|
| Y | Number of DB Write Records | The number of database writes (records/rows) in the historical logger process that is written to the database at the time the counter is polled. |

| Always ON? | Counter Name | Description |
|---|---|---|
| Y | DB Write Average Time | The average database write time expresses the average amount of time, in 100 nanosecond units, required to write data to a table in the central controller database. This value represents the average time per write of the write operations that occurred in the past second. This object is a good indicator of contention for database access. |
| Y | DB Write Records Processed | The number of records processed – written to the database – in the Historical Logger Process in the past second. |

# Administration & Data Server

**Performance Object**

Cisco ICM Distributor RealTime

**Counter Instance**

{Instance Name} ADS#

*Table 10: Administration & Data Server Real-Time Counter*

| Always ON? | Counter Name | Description |
|---|---|---|
| Y | Agent Queue Depth | The queue depth – number of pending write transactions – for the Agent table in the Real-time Client process. |
| Y | Agent DB Write Average Time | The average time – in units of 100 ns – for the Real-time Client process to write an Agent table transaction within the past 1 second interval. |
| Y | Agent DB Write Records Processed | The number of Agent table records written by the Real-time Client process in the past 1 second interval. |
| Y | Agent Skill Group Queue Depth | The queue depth – number of pending write transactions – for the Agent Skill Group table in the Real-time Client process. |
| Y | Agent Skill Group DB Write Average Time | The average time – in units of 100 ns – for the Real-time Client process to write an Agent Skill Group table transaction within the past 1 second interval. |
| Y | Agent Skill Group DB Write Records Processed | The number of Agent Skill Group table records written by the Real-time Client process in the past 1 second interval. |
| Y | Skill Group Queue Depth | The queue depth – number of pending write transactions – for the Skill Group table in the Real-time Client process. |
| Y | Skill Group DB Write Average Time | The average time – in units of 100 ns – for the Real-time Client process to write an Skill Group table transaction within the past 1 second interval. |
| Y | Skill Group DB Write Records Processed | The number of Skill Group table records written by the Real-time Client process in the past 1 second interval. |
| Y | CallType Queue Depth | The queue depth – number of pending write transactions – for the CallType table in the Real-time Client process. |
| Y | CallType DB Write Average Time | The average time – in units of 100 ns – for the Real-time Client process to write an CallType table transaction within the past 1 second interval. |

| Always ON? | Counter Name | Description |
|---|---|---|
| Y | CallType DB Write Records Processed | The number of CallType table records written by the Real-time Client process in the past 1 second interval. |
| Y | Route Queue Depth | The queue depth – number of pending write transactions – for the Route table in the Real-time Client process. |
| Y | Route DB Write Average Time | The average time – in units of 100 ns – for the Real-time Client process to write an Route table transaction within the past 1 second interval. |
| Y | Route DB Write Records Processed | The number of Route table records written by the Real-time Client process in the past 1 second interval. |
| Y | Service Queue Depth | The queue depth – number of pending write transactions – for the Service table in the Real-time Client process. |
| Y | Service DB Write Average Time | The average time – in units of 100 ns – for the Real-time Client process to write an Service table transaction within the past 1 second interval. |
| Y | Service DB Write Records Processed | The number of Service table records written by the Real-time Client process in the past 1 second interval. |

**Performance Object**
　Cisco ICM Distributor Replication
**Counter Instance**
　{Instance Name} Distributor #

**Table 11: Administration & Data Server Replication Counters**

| Always ON? | Counter Name | Description |
|---|---|---|
| Y | DB Write Average Time | The average time – in units of 100 nanoseconds – for database write operations in the HDS Replication process during the past 1 second interval. |
| Y | DB Write Records Processed | The number of records written by the HDS Replication process in the past 1 second interval. |

# PG – OPC

**Performance Object: Default**
　Cisco ICM OPC
**Optionally Enabled**
　Cisco ICM OPC (Optional)
**Counter Instance**
　"{Instance Name} PG#A/B" (For example, "acme PG3A")

**Table 12: PG - OPC Counters**

| Always ON? | Counter Name | Description |
|---|---|---|
| Y | Call Count | Number of calls that are currently active. |
| N | Agent Count | Number of agents that are configured in the system. An agent is a specific individual who receives calls through the peripheral. |

| Always ON? | Counter Name | Description |
|---|---|---|
| Y | Active Agent Answers Enabled Calls | Total number of active calls on which Agent Answers is enabled, for a given peripheral gateway. The Agent Answers service silently monitors a call and displays relevant real-time suggestions to agents and supervisors in Finesse gadgets. These suggestions enrich an agent's ability to respond to a caller. |
| N | Skill Group Count | This counter provides the number of various skill groups available for the agents to sign in. A skill group is a group of agents who share a common set of skills and who can, therefore, all handle specific types of calls. Each skill group contains one or more agents. If supported by the peripheral, each agent can be a member of more than one skill group. |
| N | Services Count | This counter provides the number of services that are configured to process the calls. A service is a type of processing that the caller requires. A peripheral might have services defined for sales, technical support, or opening new accounts. Each service has one or more skill groups whose members can provide the service. Each skill group can be associated with more than one service. |
| Y | Logged-In Agent Count | Number of agents that have logged in. This does not necessarily indicate that the agents are ready to accept calls. |
| Y | Ready Agent Count | Number of Agents that are logged in and are ready to accept calls. |
| N | Not-Ready Agent Count | Number of Agents that are logged in, but occupied with tasks other than accepting incoming calls. |
| Y | Talking Agent Count | Number of Agents currently talking on Inbound or Outbound calls. |
| N | Held Agent Count | Number of Agents that are inactively participating in a call. |
| N | Work-Ready Agent Count | Agents occupied with work associated with the last call. This implies that the agent is no longer connected to the call and is ready to receive additional calls when they exit this state. |
| N | Work-Not-Ready Agent Count | Agents occupied with work associated with the last call. This implies that the agent is no longer connected to the call. These Agents are not ready to receive additional calls when they exit this state. |
| N | Logged-Out Agent Count | Number of Agents that are logged out of the system. This count helps in validating the statistics if there are any state mismatches. |
| N | None-State Call Count | This counter provides the number of calls for which a call object was created but no activity. |
| N | Null-State Call Count | This counter provides the number of calls that has no relationship between the call and the device. |
| N | Initiated Call Count | This counter provides the number of calls for which the device has requested for a service. Often this is the dialing state. |

| Always ON? | Counter Name | Description |
|---|---|---|
| N | Alerting Call Count | This counter provides the number of calls for which the device is in alerting (ringing) state. This indicates that a call wishes to become connected to a device. |
| Y | Connected Call Count | This counter provides the number of calls for which the device is actively participating in the call. |
| N | Held Call Count | This counter provides the number of calls for which the device is inactively participating in the call. |
| N | Queued Call Count | This counter provides the number of calls for which the general state progression has been stalled. This state generally refers to two conditions but can apply to others as well. One condition is when a device is trying to establish a connection with a call, and the process is stalled. The second condition is when a call tries to establish a connection with a device and that process is stalled. |
| N | Failed Call Count | This counter provides the number of calls for which the general state progression has been closed. This state generally refers to the condition when a device tries to become connected to a call or a call tries to become connected to a device and the attempt fails. Failed can result because of failure to connect the calling device and call, failure to connect the called device and call, failure to create the call, and other reasons. |

**Enable optional counters**

**Key**
    HKEY_LOCAL_MACHINE_SOFTWARE\Cisco Systems,
    Inc.\ICM\<*Instance*>\<*PG##*>\PG\CurrentVersion\OPC
**Name**
    OPCOptionalPerfmonCounters
**Type**
    REG_DWORD
**Default**
    0
**Enabled**
    1

# PG – Communications Manager (EA) PIM

**Performance Object: Default**
    Cisco ICM CMPIM
**Optionally Enabled**
    Cisco ICM CMPIM (Optional)

**Counter Instance**

"{Instance Name} PG#A/B PIM#" (For example, "acme PG3A PIM1")

*Table 13: PG - CM PIM Counters*

| Always ON? | Counter Name | Description |
|:---:|---|---|
| N | Agent Count | Number of agents that are currently configured in system. |
| N | Calls per sec | Number of incoming calls per second. |
| Y | Call Count | Number of calls that are in progress. |
| N | Invalid Call Count | Number of calls that are not in any of the valid call states. |
| N | Messages per second | Number of call events, agent events exchanged per second between the JTAPI Gateway and CM PIM. |
| N | Messages sent | Number of call events, agent events, and CSTA messages sent today. |
| N | Messages sent past 5 | Number of call events, agent events, and CSTA messages sent past 5 seconds. |

### Enable Optional Counters

**Key**

    HKEY_LOCAL_MACHINE_SOFTWARE\Cisco Systems,
    Inc.\ICM\<Instance>\<PG##>\PG\CurrentVersion\PIMS\pim#\EAGENTData\Dynamic

**Name**

    EnableOptionalCounters

**Type**

    REG_DWORD

**Default**

    0

**Enabled**

    1

# PG – VRU PIM

**Performance Object**

Cisco ICM VRUPIM

**Counter Instance**

"{Instance Name} PG#A/B PIM#" (For example, "acme PG3A PIM3")

*Table 14: PG - VRU PIM Counters*

| Always ON? | Counter Name | Description |
|:---:|---|---|
| Y | Calls At VRU | Calls at VRU is the number of calls that are currently at the Voice Response Unit (VRU). For a VRU that only uses a Call Routing Interface, this value is zero. |
| N | Messages To VRU/sec | Messages To VRU/sec is the rate at which messages are sent to the Voice Response Unit (VRU). This counter is active only when enabled in ICM registry. |
| N | Messages From VRU/sec | Messages From VRU/sec is the rate at which messages are received from the Voice Response Unit (VRU). This counter is active only when enabled in ICM registry. |

| Always ON? | Counter Name | Description |
|---|---|---|
| N | Bytes To VRU/sec | Bytes To VRU/sec is the rate at which bytes are sent to the Voice Response Unit (VRU). This counter is active only when enabled in ICM registry. |
| N | Bytes From VRU/sec | Bytes From VRU/sec is the rate at which bytes are received from the Voice Response Unit (VRU). This counter is active only when enabled in ICM registry. |
| Y | New Calls/sec | New Calls/sec is the rate at which new calls arriving at the Voice Response Unit (VRU). New calls are calls not under ICM script control when arriving at a Service Control VRU. The Pre-Routed calls/Sec can be for an entire day. The calls/sec can also be from the time the PIM restarts or for an hour. |
| Y | Pre-Routed Calls/Sec | Pre-Routed Calls/sec is the rate at which Pre-Routed calls are arriving at Voice Response Unit (VRU). Pre-Routed calls are calls under ICM script control when arriving at a Service Control VRU. The Pre-Routed calls/Sec can be for an entire day. The calls/sec can also be from the time the PIM restarts or for an hour. |
| Y | Connection Resets | Connection Resets is the number of times the TCP connection between ICM and the Voice Response Unit changed from an established state to a closed state since the application started. |

### Enable Optional Counters

**Key**

> HKEY_LOCAL_MACHINE_SOFTWARE\Cisco Systems,
> Inc.\ICM\*<Instance>*\*<PG##>*\PG\CurrentVersion\PIMS\pim#\VRUData\Dynamic

**Name**

> EnableOptionalPerfmonCounter

**Type**

> REG_DWORD

**Default**

> 0

**Enabled**

> 1

# CTI Server

**Performance Object: Default**

> Cisco ICM CTISVR

**Optionally Enabled**

> Cisco ICM CTISVR (Optional)

**Counter Instance**

"{Instance Name} CG#A/B" (For example, "acme CG3A")

*Table 15: CTI Server Counters*

| Always ON? | Counter Name | Description |
|---|---|---|
| N | Reported Call Count | Number of calls that are already reported to the CTI clients. |
| N | Active Call Count | Number of calls that are currently in progress. |
| N | Deactivated Call Count | Number of calls that are not currently active and eventually cleared. |
| N | Cleared Call Count | Number of calls that no longer exist in the system. |
| N | Private Call Count | Number of calls that are privately tracked by CTI Server and which are not reported to OPC. |
| Y | Logged-In Agent Count | Agents that have logged in. This does not necessarily indicate that they are ready to accept calls. |
| Y | Ready Agent Count | Number of Agents that are logged in and are ready to accept calls. |
| N | Not-Ready Agent Count | Number of Agents that are logged in, but occupied with tasks other than accepting incoming calls. |
| Y | Talking Agent Count | Number of Agents currently talking on Inbound or Outbound calls. |
| N | Held Agent Count | Number of Agents that are inactively participating in a call. |
| N | Work-Ready Agent Count | Agents occupied with work associated with the last call. This implies that agent is no longer connected to the call and is ready to receive additional calls when they exit this state. |
| N | Work-Not-Ready Agent Count | Agents occupied with work associated with the last call. This implies that agent is no longer connected to the call. These agents are not ready to receive additional calls when they exit this state. |
| N | Logged-Out Agent Count | The number of Agents that are logged out of the system. This count helps in validating the statistics if there are any state mismatches. |
| Y | Sessions Unknown | The number of sessions for which there is no socket connection made yet. |
| N | Sessions Opening | The number of sessions that are in the process of setting up a connection. |
| Y | Sessions Open | The number of sessions that were successfully setup. |
| N | Sessions Closing | The number of sessions that are in the process of tear down. |
| Y | Sessions Closed | The total number of sessions that are terminated by the CTI Server. |
| Y | Sessions Failed | The number of sessions that failed due to various reasons like missing heartbeat, open request timeout, session inactivity, and so on. These timers are configurable parameters in CTI Server. |
| Y | Total Sessions | The total number of sessions maintained by CTI Server. |

**Enable Optional Counters**

**Key**
```
HKEY_LOCAL_MACHINE_SOFTWARE\Cisco Systems,
Inc.\ICM\<Instance>\<CG##>\CG\CurrentVersion\CTIServer\Dynamic
```
**Name**
CTISVROptionalCounters
**Type**
REG_DWORD
**Default**
0
**Enabled**
1

# CTI OS Server

**Performance Object**
Cisco ICM CTI OS
**Counter Instance**
CTI OS Name

*Table 16: CTI OS Server Counters*

| Always ON? | Counter Name | Description |
|---|---|---|
| Y | CTI OS Active Client Connections | The number of CTI OS Active Client Mode Desktop Connections. This value indicates the total number of desktops connected to the CTI OS server. The number of desktops connected to the A and B side of CTI OS determine the total desktops connected through this instance of CTI OS server. |
| Y | CTI OS Active Monitor Mode Connections | The number of CTI OS Active Monitor Mode Desktop Connections. CTI OS only supports two monitor mode connections per each CTI OS server. This value indicates how many monitor mode connections are in use. After there are two in use further monitor mode connection attempts are rejected. |
| Y | CTI OS Active Calls | The total number of active calls being tracked by CTI OS. This value shows how many calls are currently being handled by CTI OS. This value should go up and down based on the call arrival rate and the agent call completion rate. |
| Y | CTI OS Configured Skill Groups | The total number of configured skill groups being tracked by CTI OS. This value should match the number of skill groups configured for the PG that this CTI OS is associated. |
| Y | CTI OS Configured Teams | The total number of configured Teams being tracked by CTI OS. This value should match the number of teams configured for the PG that this CTI OS is associated. |
| Y | CTI OS Configured Agents | The total number of configured Agents being tracked by CTI OS. This value should match the number of Agents configured for the PG that this CTI OS is associated. |
| Y | CTI OS Active Conferences | The total number of active Conferences being tracked by CTI OS. This value indicates the number of multi-party calls that are in progress at any one given time in CTI OS. |

| Always ON? | Counter Name | Description |
|---|---|---|
| Y | CTI OS Call Count | The total number of calls handled by CTI OS. This value only increases and shows the total number of calls processed by CTI OS since it last started. This value should increase at the same rate as the calls per second being shown by the Router. |
| Y | CTI OS Conference Count | The total number of Conferences performed by CTI OS. This value only increases and shows the total number of calls that were conferenced since CTI OS last started. The conference count should be a small percentage of total calls. |
| Y | CTI OS Transfer Count | The total number of Transfers performed by CTI OS. This value only increases and shows the total number of calls that were transferred since CTI OS last started. The transfer count should be a small percentage of total calls. |
| Y | CTI OS Call Failed Count | The total number of Calls that failed reported to CTI OS. This value shows the total number of calls that failed via a failure event being reported to CTI OS. If this count begins to rise the log file should be captured to gather more specific information about the failure events. |
| Y | CTI OS CTI Message Receive Rate (msg/sec) | The rate at which CTI OS receives messages from CTI Server per second. This value is an indicator to total load on the system. Increases are not really a problem unless the CTI OS Service Broker Queue Size also begins to increase. |
| Y | CTI OS CTI Message Send Rate (msg/sec) | The rate at which CTI OS sends messages to CTI Server per second. This value is an indicator of total load on the system. If it increases it indicate the CTI OS server is under a heavy request load from the desktop clients. |
| Y | CTI OS Service Broker Queue Size | The number of messages queued in the CTI OS Service Broker queue. This value is a good load indicator for CTI OS. If it increases, it can indicate that CTI OS is not keeping up with the incoming message rate from CTI Server. A review of the configuration may be necessary to understand why CTI OS is not able to keep up with event handling from CTI Server. |
| N | CTI OS Call Object Count | The total number of CTI OS call objects that are active. This value shows how many CTI OS Call objects were created since it last started. This value should go up and down and may reach a steady state when the number of calls being completed by agents equals the call arrival rate. |
| N | CTI OS Connection Object Count | The total number of active CTI OS connection objects. This value shows how many CTI OS connection objects were created since it last started. This value should go up and down and may reach a steady state when the number of calls being completed by agents equals the call arrival rate. |
| N | CTI OS Argument Object Count | The total number of active CTI OS argument objects. This value shows how many CTI OS argument objects were created since it last started. This value shall be quite large, go up and down and may reach a steady state when the number of calls being completed by agents equals the call arrival rate. |
| N | CTI OS Device Object Count | The total number of active CTI OS devices. This value shows how many CTI OS device objects were created since it last started. This value should mainly stay constant while CTI OS runs. |
| N | CTI OS Agent Object Count | The total number of CTI OS agent objects. This value shows how many CTI OS agent objects were created since it last started. This value should stay constant while CTI OS runs unless agents are added or deleted. |

| Always ON? | Counter Name | Description |
|:---:|---|---|
| N | CTI OS Skill group Object Count | The total number of CTI OS skill group objects. This value shows how many CTI OS skill group objects were created since it last started. This value should stay constant while CTI OS runs unless skill groups are added or deleted. |
| N | CTI OS Supervisor Object Count | The total number of CTI OS Supervisor objects. This value shows how many CTI OS supervisor objects were created since it last started. This value should stay constant while CTI OS runs unless supervisors are added or deleted. |
| N | CTI OS Team Object Count | The total number of CTI OS Team objects. This value shows how many CTI OS team objects were created since it last started. This value stays constant while CTI OS runs unless teams are added or deleted. |
| N | CTI OS Total Objects Created Count | The total count of all objects created by CTI OS. This value shows how many CTI OS objects were created since it last started. This value only increases and grows very large as CTI OS up time increases. |
| N | CTI OS Total Objects Deletion Count | The total count of all objects deleted by CTI OS. This value shows how many CTI OS objects were deleted since it last started. This value only increases and grows very large as CTI OS up time increases. It never equals the total objects created count as some objects are never deleted after being created by CTI OS like agent, device, team and skill group objects. |
| N | CTI OS Active Object Count | The total count of all objects created by CTI OS that are active. This value shows how many CTI OS objects are currently allocated since it last started. If this value begins to increase it would indicate that a memory leak is occurring in CTI OS. The specific object counters show which object is not being released. |
| Y | CTI OS CLIENT Send Message Rate (msg/sec) | The rate at which CTI OS sends messages to Clients per second. This value shows the number of messages, per second, that CTI OS is delivering messages to CTI OS desktops. As this value increases it indicates that CTI OS server is being placed under an increasing load. A review of the configuration as it relates to agents, skill groups and teams may be necessary. |
| Y | CTI OS CLIENT Receive Message Rate (msg/sec) | The rate at which CTI OS receives messages from Clients per second. This value shows the number of messages, per second, that are being received from the CTI OS desktops. As this value increases it indicates that CTI OS is being placed under an increasing request load from the desktops. |
| Y | CTI OS CLIENT Total Number of Pending Write Operations | The total number of pending write operations for all clients. This value shows the total number of messages in the system waiting to be read by CTI OS clients. If the value increases, it can indicate that there are one or more clients not keeping up with reading messages from CTI OS. |
| Y | CTI OS CLIENT Total Message Buffer Size (Bytes) | The total number of bytes used to store the pending writes for all clients. This value shows the total amount of memory used to store all the messages that are waiting to be read by CTI OS clients. |
| Y | CTI OS CG Receive Queue Size | The number of messages queued in the CTI OS CG Receive Queue. This value is an indicator of the total load on the system. If it increases, a review of the configuration may be necessary to understand why CTI OS is not keeping up with the incoming message rate from the CTI Server. |

**Enable Optional Counters**

**Key**

```
HKEY_LOCAL_MACHINE_SOFTWARE\Cisco Systems,
Inc.\ICM\{instance}\CTIOS#\EMS\CurrentVersion\Library\Processes\ctios
```

**Name**

EMSTraceMask

**Type**

REG_DWORD

**Enable**

0x200000

# Outbound Option Campaign Manager

**Performance Object**

Cisco ICM CampaignMgr

**Counter Instance**

"{Instance Name}"

*Table 17: Outbound Option Campaign Manager Counters*

| Always ON? | Counter Name | Description |
|---|---|---|
| Y | DB Space Utilization | The Campaign Manager and Import processes share a private database on the Logger. This counter reports the percentage of allocated space in the database that is currently utilized. An administrator should consider increasing the database size when the value of this counter exceeds eighty percent (80%). |
| Y | Queue Depth | The Campaign Manager is a multithreaded process. There is one main dispatch thread that is involved in most processing. Queue Depth indicates how many messages are queued to this internal dispatch thread. |
| Y | Average Queue Time | The Campaign Manager is a multithreaded process. There is one main dispatch thread that is involved in most processing. This counter reports the average time spent in the main dispatch thread queue in milliseconds. |
| Y | Do Not Call Number Count | The Campaign Manager manages a global list of phone numbers used to prevent block dialing. This list is stored in memory. Each record uses 17 bytes of memory. This counter shows how many do not call entries are currently in memory. |
| Y | Active Dialer Count | The Campaign Manager process feeds one or more Dialer components; each Dialer dials customer numbers for outbound campaigns. This counter indicates how many Dialers are currently registered to the Campaign Manager. |

| Always ON? | Counter Name | Description |
|---|---|---|
| Y | Congestion Level | This counter returns the current Campaign Manager congestion level. The Campaign Manager becomes congested when it cannot process the volume of inbound messages from dialers quickly enough. Congestion control engages when the Campaign Manager message queue depth reaches predefined thresholds. As each level is reached, the Campaign Manager instructs dialers to slow the dialing rate to a sustainable level. When congestion eases to a lower level, the Campaign Manager signals for an increased dialing rate. The congestion levels are:<br><br>• 0 - Normal operation<br><br>• 1 - Slightly congested<br><br>• 2 - Moderately congested<br><br>• 3 - Heavily congested |
| Y | Replication Pending Files | This counter returns the number of pending files that await replication from this side. |
| Y | Replication Pending Kilo Bytes | This counter returns the amount of data (in KB) that await replication from this side. |

# Outbound Option Import

**Performance Object**
    Cisco ICM Import
**Counter Instance**
    "{Instance Name}"

*Table 18: Outbound Option Import Counters*

| Always ON? | Counter Name | Description |
|---|---|---|
| Y | Records Imported Today | The Outbound Option Import process imports customer records that contain phone numbers used by the Campaign Manager and Dialer to find available customers for a campaign. This counter tracks how many records were imported today. |

# Outbound Option Dialer

**Performance Object**
    Cisco ICM Dialer

## Counter Instance

"{Instance Name}"

*Table 19: Outbound Option Dialer Counters*

| Always ON? | Counter Name | Description |
|:---:|---|---|
| Y | Queue Depth | The Dialer is a multithreaded process that communicates between threads using inter thread messaging. This indicates how many messages are currently queued up for the main dispatch thread. By default, the Dialer process restarts when this value exceeds 10,000 messages. |
| Y | Average Queue Time | The Dialer is a multithreaded process that communicates between threads using messaging. There is one main dispatch thread that is involved in most processing. This shows what is the average time spent in queue. |
| Y | Talking Agents | For an agent campaign, the Dialer replaces calls to customers and transfers those customers to agents. This counter indicates how many agents are currently talking in the monitored campaign skill group. |
| Y | Busy Port (Customer) Count | The port is the unit on the Dialer that places calls to reserve agents and to contact customers. This counter tracks how many ports are currently busy trying to contact customers. This includes ports that are actively dialing and those that have been allocated but are not yet dialing a customer. |
| Y | Ports Actively Dialing Customer Count | The port is the unit on the Dialer that places calls to reserve agents and to contact customers. This counter tracks ports that are currently actively dialing customers. |
| Y | Busy Port (Reservation) Count | The port is the unit on the Dialer that places calls to reserve agents and to contact customers. This counter tracks how many ports are currently busy reserving agents only for normal records. |
| Y | Agent Reservation Port Count | The port is the unit on the Dialer that places calls to reserve agents and to contact customers. This counter tracks how many ports are currently being used for reserving agents for both normal and callback records. |
| Y | Port Utilization Percent | The port is the unit on the Dialer that places calls to reserve agents and to contact customers. This counter tracks the percent of total dialer ports in use for all outbound calls. |
| Y | Idle Port Count | The port is the unit on the Dialer that places calls to reserve agents and to contact customers. This counter tracks how many ports are currently idle. |
| Y | PCB Records Cached | The Dialer caches the PCB records received from campaign manager. This counter tracks the total number of PCB records cached. |
| Y | Idle PCB Records Cached | This counter tracks the number of the cached PCB records that are in the idle state, waiting to be dialed out from the dialer. |
| Y | Idle Not-Ready PCB Records Cached | This counter tracks the number of cached PCB records in the idle state where Agent is in Not-Ready state. These records will be retried until the agent is ready or the PCB call time expires. |
| Y | Call Attempt Count | The Dialer attempts to contact customers and transfer them to reserved agents or an available IVR. This counter tracks how many customer attempts were placed today. It does not include preview calls that were rejected or skipped. |

| Always ON? | Counter Name | Description |
|---|---|---|
| Y | Abandoned Call Count | When a customer is contacted and an agent is not available to take the call, the call can be dropped or sent to the IVR for prompting and queuing. When either of these conditions occurs, the all is counted as abandoned. In a transfer to IVR campaign, a call is dropped and counted as abandoned if the configured IVR port limit is exceeded. |
| Y | Reservation Call Count | The Dialer places calls to agents to reserve them for use while attempting to contact available customers. This counter tracks how many reservation calls were placed today. |
| Y | Answering Machine Call Count | A campaign can be enabled to differentiate between live voice and answering machines. This counter tracks how many answering machines were detected today. |
| Y | Customer Answered Call Count | A campaign can be enabled to differentiate between live voice and answering machines. If answering machine detection (AMD) is enabled for a campaign this counter increments when live voice is detected. If AMD is disabled, then all connected calls that are not FAX are identified as live voice. Direct Preview calls are identified as voice or AMD by the agent. This counter is reset daily at midnight. |
| Y | Customer Not Answered Call Count | The Dialer attempts to contact customers. This counter tracks how many attempts resulted in no answer condition. This counter is reset daily. |
| Y | Error Call Count | The Dialer attempts to contact customers. This counter tracks how many attempts resulted in a network error condition which includes no ring-back, no dial tone, and call disconnected from the network before ring no answer time out was exceeded. |
| Y | Number of attempted calls per second | This counter tracks how many calls per second the Dialer is placing rounded to the nearest integer. If the dialing rate is too high, it can result in network congestion on the voice network that can result in inefficient dialing. |

# Message Delivery Service

**Performance Object**
Cisco ICM MDSCLIENT
**Counter Instance**
"{Instance Name}"

*Table 20: MDS Client Counters*

| Always ON? | Counter Name | Description |
|---|---|---|
| N | Client Handle ID | Handle for this MDS client. It is used to uniquely identify the MDS client connected to the MDS process. |
| N | Now Message Received | Number of messages received by the MDS client per second. |
| N | Now Message Sent | Number of messages sent by the MDS client per second. |
| N | Now Bytes Received | Number of bytes received by the MDS client per second. |
| N | Now Bytes Sent | Number of bytes sent by the MDS client per second |

| Always ON? | Counter Name | Description |
|---|---|---|
| N | Current Buffers Memory Allocated | Total number of bytes used by all currently allocated buffers. |
| N | Current Buffers Allocated | Total number of buffers currently allocated from buffer pool. |
| N | Buffers Allocation Requests/sec | Number of buffers allocated per second. |
| N | Buffers Free Requests/sec | Number of buffers freed per second. |
| N | Current Buffers Memory Limit | Maximum amount of memory allowed to be allocated for buffers for this process. |
| N | Initial Buffers Memory Limit | Amount of memory limit reserved for buffers for this process. |
| N | SendClientQ Current Depth | Current number of messages in the MDS Client Send Queue. |
| N | SendClientQ Now Messages In/sec | Total number of messages added to the MDS Client Send Queue per second. |
| N | SendClientQ Now Messages Out/sec | Total number of messages removed from the MDS Client Send Queue per second. |
| N | SendClientQ Now Bytes In/sec | Total number of bytes added for all messages to the MDS Client Send Queue per second. |
| N | SendClientQ Now Bytes Out/sec | Total number of bytes removed for all the messages from the MDS Client Send Queue per second. |
| N | SendClientQ Now Traffic Intensity | Ratio of the number of messages added to the number of messages removed from the MDS Client Send Queue per second. |
| N | SendClientQ Avg. Queue Response Time [ms] | Average time in milliseconds a message waits in the MDS Client Send Queue. |
| N | SendClientQ 90% Queue Response Time [ms] | The response time in milliseconds that 90% of all messages passing through the MDS Client Send Queue experience. |

**Performance Object**
Cisco ICM MDSPROCCLIENT
**Counter Instance**
"{Instance name}"

*Table 21: MDS Process Client Counters*

| Always ON? | Counter Name | Description |
|---|---|---|
| N | Client Handle ID | Handle for this MDS client. It is used to uniquely identify the MDS client connected to the MDS process. |
| N | Total MDS Client Connects | Total number of times the MDS client has connected to the MDS process. |
| N | Total MDS Client Disconnects | Total number of times the MDS client has disconnected from the MDS process. |
| N | Now Message Received from Client | Number of messages received from the MDS client per second. |

| Always ON? | Counter Name | Description |
|:---:|---|---|
| N | Now Message Sent to Client | Number of messages sent to the MDS client per second. |
| N | Now Bytes Received from Client | Number of bytes received from the MDS client per second. |
| N | Now Bytes Sent to Client | Number of bytes sent to the MDS client per second. |
| N | ToClientQ Current Depth | Current number of messages in the MDS Send Client Queue. |
| N | ToClientQ Now Messages In/sec | Total number of messages added to the MDS Client Send Queue per second. |
| N | ToClientQ Now Messages Out/sec | Total number of messages removed from the MDS Client Send Queue per second. |
| N | ToClientQ Now Bytes In/sec | Total number of bytes added for all messages to the MDS Client Send Queue per second. |
| N | ToClientQ Now Bytes Out/sec | Total number of bytes removed for all the messages from the MDS Client Send Queue per second. |
| N | ToClientQ Now Traffic Intensity | Ratio of the number of messages added to the number of messages removed from the MDS Client Send Queue per second. |
| N | ToClientQ Avg. Queue Response Time [ms] | Average time in milliseconds a message waits in the MDS Client Send Queue. |
| N | ToClientQ 90% Queue Response Time [ms] | The response time in milliseconds that 90% of all messages passing through the MDS Client Send Queue experience. |

**Performance Object**

Cisco ICM MDSPROC

**Counter Instance**

"{Instance Name}"

**Table 22: MDS Process Counters**

| Always ON? | Counter Name | Description |
|:---:|---|---|
| N | Current Buffers Memory Allocated | Total number of bytes used by all currently allocated buffers. |
| N | Current Buffers Allocated | Total number of buffers currently allocated from buffer pool. |
| N | Buffers Allocation Requests/sec | Number of buffers allocated per second. |
| N | Buffers Free Requests/sec | Number of buffers freed per second. |
| N | Current Buffers Memory Limit | Maximum amount of memory allowed to be allocated for buffers for this process. |
| N | Initial Buffers Memory Limit | Amount of memory limit reserved for buffers for this process. |
| N | Synch Messages Ordered/sec | Number of messages ordered by the MDS synchronizer per second. |

| Always ON? | Counter Name | Description |
|---|---|---|
| N | Synch MDS Duplicates/sec | Number of duplicate MDS messages detected by the synchronizer per second. |
| N | Synch DMP Duplicates/sec | Number of duplicate DMP messages detected by the synchronizer per second. |
| N | LocalHighInQ Current Depth | Current number of messages in the Local High Incoming Queue. |
| N | LocalHighInQ Now Messages In/sec | Total number of messages added to the Local High Incoming Queue per second. |
| N | LocalHighInQ Now Messages Out/sec | Total number of messages removed from the Local High Incoming Queue per second. |
| N | LocalHighInQ Now Bytes In/sec | Total number of bytes added for all messages to the Local High Incoming Queue per second. |
| N | LocalHighInQ Now Bytes Out/sec | Total number of bytes removed for all the messages from the Local High Incoming Queue per second. |
| N | LocalHighInQ Now Traffic Intensity | Ratio of the number of messages added to the number of messages removed from the Local High Incoming Queue per second. |
| N | LocalHighInQ Avg. Queue Response Time [ms] | Average time in milliseconds a message waits in the Local High Incoming Queue. |
| N | LocalHighInQ 90% Queue Response Time [ms] | The response time in milliseconds that 90% of all messages passing through the Local High Incoming Queue experience. |
| N | LocalMedInQ Current Depth | Current number of messages in the Local Medium Incoming Queue. |
| N | LocalMedInQ Now Messages In/sec | Total number of messages added to the Local Medium Incoming Queue per second. |
| N | LocalMedInQ Now Messages Out/sec | Total number of messages removed from the Local Medium Incoming Queue per second. |
| N | LocalMedInQ Now Bytes In/sec | Total number of bytes added for all messages to the Local Medium Incoming Queue per second. |
| N | LocalMedInQ Now Bytes Out/sec | Total number of bytes removed for all the messages from the Local Medium Incoming Queue per second. |
| N | LocalMedInQ Now Traffic Intensity | Ratio of the number of messages added to the number of messages removed from the Local Medium Incoming Queue per second. |
| N | LocalMedInQ Avg. Queue Response Time [ms] | Average time in milliseconds a message waits in the Local Medium Incoming Queue. |
| N | LocalMedInQ 90% Queue Response Time [ms] | The response time in milliseconds that 90% of all messages passing through the Local Medium Incoming Queue experience. |

| Always ON? | Counter Name | Description |
|---|---|---|
| N | LocalLowInQ Current Depth | Current number of messages in the Local Low Incoming Queue. |
| N | LocalLowInQ Now Messages In/sec | Total number of messages added to the Local Low Incoming Queue per second. |
| N | LocalLowInQ Now Messages Out/sec | Total number of messages removed from the Local Low Incoming Queue per second. |
| N | LocalLowInQ Now Bytes In/sec | Total number of bytes added for all messages to the Local Low Incoming Queue per second. |
| N | LocalLowInQ Now Bytes Out/sec | Total number of bytes removed for all the messages from the Local Low Incoming Queue per second. |
| N | LocalLowInQ Now Traffic Intensity | Ratio of the number of messages added to the number of messages removed from the Local Low Incoming Queue per second. |
| N | LocalLowInQ Avg. Queue Response Time [ms] | Average time in milliseconds a message waits in the Local Low Incoming Queue. |
| N | LocalLowInQ 90% Queue Response Time [ms] | The response time in milliseconds that 90% of all messages passing through the Local Low Incoming Queue experience. |
| N | RemoteHighOutQ Current Depth | Current number of messages in the Remote High Output Queue. |
| N | RemoteHighOutQ Now Messages In/sec | Total number of messages added to the Remote High Output Queue per second. |
| N | RemoteHighOutQ Now Messages Out/sec | Total number of messages removed from the Remote High Output Queue per second. |
| N | RemoteHighOutQ Now Bytes In/sec | Total number of bytes added for all messages to the Remote High Output Queue per second. |
| N | RemoteHighOutQ Now Bytes Out/sec | Total number of bytes removed for all the messages from the Remote High Output Queue per second. |
| N | RemoteHighOutQ Now Traffic Intensity | Ratio of the number of messages added to the number of messages removed from the Remote High Output Queue per second. |
| N | RemoteHighOutQ Avg. Queue Response Time [ms] | Average time in milliseconds a message waits in the Remote High Output Queue. |
| N | RemoteHighOutQ 90% Queue Response Time [ms] | The response time in milliseconds that 90% of all messages passing through the Remote High Output Queue experience. |
| N | RemoteMedOutQ Current Depth | Current number of messages in the Remote Medium Output Queue. |
| N | RemoteMedOutQ Now Messages In/sec | Total number of messages added to the Remote Medium Output Queue per second. |

| Always ON? | Counter Name | Description |
|:---:|---|---|
| N | RemoteMedOutQ Now Messages Out/sec | Total number of messages removed from the Remote Medium Output Queue per second. |
| N | RemoteMedOutQ Now Bytes In/sec | Total number of bytes added for all messages to the Remote Medium Output Queue per second. |
| N | RemoteMedOutQ Now Bytes Out/sec | Total number of bytes removed for all the messages from the Remote Medium Output Queue per second. |
| N | RemoteMedOutQ Now Traffic Intensity | Ratio of the number of messages added to the number of messages removed from the Remote Medium Output Queue per second. |
| N | RemoteMedOutQ Avg. Queue Response Time [ms] | Average time in milliseconds a message waits in the Remote Medium Output Queue. |
| N | RemoteMedOutQ 90% Queue Response Time [ms] | The response time in milliseconds that 90% of all messages passing through the Remote Medium Output Queue experience. |
| N | RemoteLowOutQ Current Depth | Current number of messages in the Remote Low Output Queue. |
| N | RemoteLowOutQ Now Messages In/sec | Total number of messages added to the Remote Low Output Queue per second. |
| N | RemoteLowOutQ Now Messages Out/sec | Total number of messages removed from the Remote Low Output Queue per second. |
| N | RemoteLowOutQ Now Bytes In/sec | Total number of bytes added for all messages to the Remote Low Output Queue per second. |
| N | RemoteLowOutQ Now Bytes Out/sec | Total number of bytes removed for all the messages from the Remote Low Output Queue per second. |
| N | RemoteLowOutQ Now Traffic Intensity | Ratio of the number of messages added to the number of messages removed from the Remote Low Output Queue per second. |
| N | RemoteLowOutQ Avg. Queue Response Time [ms] | Average time in milliseconds a message waits in the Remote Low Output Queue. |
| N | RemoteLowOutQ 90% Queue Response Time [ms] | The response time in milliseconds that 90% of all messages passing through the Remote Low Output Queue experience. |
| N | LocalHighOrderQ Current Depth | Current number of messages in the Local High Order Queue. |
| N | LocalHighOrderQ Now Messages In/sec | Total number of messages added to the Local High Order Queue per second. |
| N | LocalHighOrderQ Now Messages Out/sec | Total number of messages removed from the Local High Order Queue per second. |
| N | LocalHighOrderQ Now Bytes In/sec | Total number of bytes added for all messages to the Local High Order Queue per second. |

| Always ON? | Counter Name | Description |
|---|---|---|
| N | LocalHighOrderQ Now Bytes Out/sec | Total number of bytes removed for all the messages from the Local High Order Queue per second. |
| N | LocalHighOrderQ Now Traffic Intensity | Ratio of the number of messages added to the number of messages removed from the Local High Order Queue per second. |
| N | LocalHighOrderQ Avg. Queue Response Time [ms] | Average time in milliseconds a message waits in the Local High Order Queue. |
| N | LocalHighOrderQ 90% Queue Response Time [ms] | The response time in milliseconds that 90% of all messages passing through the Local High Order Queue experience. |
| N | LocalMedOrderQ Current Depth | Current number of messages in the Local Medium Order Queue. |
| N | LocalMedOrderQ Now Messages In/sec | Total number of messages added to the Local Medium Order Queue per second. |
| N | LocalMedOrderQ Now Messages Out/sec | Total number of messages removed from the Local Medium Order Queue per second. |
| N | LocalMedOrderQ Now Bytes In/sec | Total number of bytes added for all messages to the Local Medium Order Queue per second. |
| N | LocalMedOrderQ Now Bytes Out/sec | Total number of bytes removed for all the messages from the Local Medium Order Queue per second. |
| N | LocalMedOrderQ Now Traffic Intensity | Ratio of the number of messages added to the number of messages removed from the Local Medium Order Queue per second. |
| N | LocalMedOrderQ Avg. Queue Response Time [ms] | Average time in milliseconds a message waits in the Local Medium Order Queue. |
| N | LocalMedOrderQ 90% Queue Response Time [ms] | The response time in milliseconds that 90% of all messages passing through the Local Medium Order Queue experience. |
| N | LocalLowOrderQ Current Depth | Current number of messages in the Local Low Order Queue. |
| N | LocalLowOrderQ Now Messages In/sec | Total number of messages added to the Local Low Order Queue per second. |
| N | LocalLowOrderQ Now Messages Out/sec | Total number of messages removed from the Local Low Order Queue per second. |
| N | LocalLowOrderQ Now Bytes In/sec | Total number of bytes added for all messages to the Local Low Order Queue per second. |
| N | LocalLowOrderQ Now Bytes Out/sec | Total number of bytes removed for all the messages from the Local Low Order Queue per second. |
| N | LocalLowOrderQ Now Traffic Intensity | Ratio of the number of messages added to the number of messages removed from the Local Low Order Queue per second. |

| Always ON? | Counter Name | Description |
|---|---|---|
| N | LocalLowOrderQ Avg. Queue Response Time [ms] | Average time in milliseconds a message waits in the Local Low Order Queue. |
| N | LocalLowOrderQ 90% Queue Response Time [ms] | The response time in milliseconds that 90% of all messages passing through the Local Low Order Queue experience. |
| N | RemoteHighOrderQ Current Depth | Current number of messages in the Remote High Order Queue. |
| N | RemoteHighOrderQ Now Messages In/sec | Total number of messages added to the Remote High Order Queue per second. |
| N | RemoteHighOrderQ Now Messages Out/sec | Total number of messages removed from the Remote High Order Queue per second. |
| N | RemoteHighOrderQ Now Bytes In/sec | Total number of bytes added for all messages to the Remote High Order Queue per second. |
| N | RemoteHighOrderQ Now Bytes Out/sec | Total number of bytes removed for all the messages from the Remote High Order Queue per second. |
| N | RemoteHighOrderQ Now Traffic Intensity | Ratio of the number of messages added to the number of messages removed from the Remote High Order Queue per second. |
| N | RemoteHighOrderQ Avg. Queue Response Time [ms] | Average time in milliseconds a message waits in the Remote High Order Queue. |
| N | RemoteHighOrderQ 90% Queue Response Time [ms] | The response time in milliseconds that 90% of all messages passing through the Remote High Order Queue experience. |
| N | RemoteMedOrderQ Current Depth | Current number of messages in the Remote Medium Order Queue. |
| N | RemoteMedOrderQ Now Messages In/sec | Total number of messages added to the Remote Medium Order Queue per second. |
| N | RemoteMedOrderQ Now Messages Out/sec | Total number of messages removed from the Remote Medium Order Queue per second. |
| N | RemoteMedOrderQ Now Bytes In/sec | Total number of bytes added for all messages to the Remote Medium Order Queue per second. |
| N | RemoteMedOrderQ Now Bytes Out/sec | Total number of bytes removed for all the messages from the Remote Medium Order Queue per second. |
| N | RemoteMedOrderQ Now Traffic Intensity | Ratio of the number of messages added to the number of messages removed from the Remote Medium Order Queue per second. |
| N | RemoteMedOrderQ Avg. Queue Response Time [ms] | Average time in milliseconds a message waits in the Remote Medium Order Queue. |

| Always ON? | Counter Name | Description |
|---|---|---|
| N | RemoteMedOrderQ 90% Queue Response Time [ms] | The response time in milliseconds that 90% of all messages passing through the Remote Medium Order Queue experience. |
| N | RemoteLowOrderQ Current Depth | Current number of messages in the Remote Low Order Queue. |
| N | RemoteLowOrderQ Now Messages In/sec | Total number of messages added to the Remote Low Order Queue per second. |
| N | RemoteLowOrderQ Now Messages Out/sec | Total number of messages removed from the Remote Low Order Queue per second. |
| N | RemoteLowOrderQ Now Bytes In/sec | Total number of bytes added for all messages to the Remote Low Order Queue per second. |
| N | RemoteLowOrderQ Now Bytes Out/sec | Total number of bytes removed for all the messages from the Remote Low Order Queue per second. |
| N | RemoteLowOrderQ Now Traffic Intensity | Ratio of the number of messages added to the number of messages removed from the Remote Low Order Queue per second. |
| N | RemoteLowOrderQ Avg. Queue Response Time [ms] | Average time in milliseconds a message waits in the Remote Low Order Queue. |
| N | RemoteLowOrderQ 90% Queue Response Time [ms] | The response time in milliseconds that 90% of all messages passing through the Remote Low Order Queue experience. |
| N | TDHighQ Current Depth | Current number of messages in the Timed Delivery High Queue. |
| N | TDHighQ Now Messages In/sec | Total number of messages added to the Timed Delivery High Queue per second. |
| N | TDHighQ Now Messages Out/sec | Total number of messages removed from the Timed Delivery High Queue per second. |
| N | TDHighQ Now Bytes In/sec | Total number of bytes added for all messages to the Timed Delivery High Queue per second. |
| N | TDHighQ Now Bytes Out/sec | Total number of bytes removed for all the messages from the Timed Delivery High Queue per second. |
| N | TDHighQ Now Traffic Intensity | Ratio of the number of messages added to the number of messages removed from the Timed Delivery High Queue per second. |
| N | TDHighQ Avg. Queue Response Time [ms] | Average time in milliseconds a message waits in the Timed Delivery High Queue. |
| N | TDHighQ 90% Queue Response Time [ms] | The response time in milliseconds that 90% of all messages passing through the Timed Delivery High Queue experience. |
| N | TDMedQ Current Depth | Current number of messages in the Timed Delivery Medium Queue. |
| N | TDMedQ Now Messages In/sec | Total number of messages added to the Timed Delivery Medium Queue per second. |

| Always ON? | Counter Name | Description |
|---|---|---|
| N | TDMedQ Now Messages Out/sec | Total number of messages removed from the Timed Delivery Medium Queue per second. |
| N | TDMedQ Now Bytes In/sec | Total number of bytes added for all messages to the Timed Delivery Medium Queue per second. |
| N | TDMedQ Now Bytes Out/sec | Total number of bytes removed for all the messages from the Timed Delivery Medium Queue per second. |
| N | TDMedQ Now Traffic Intensity | Ratio of the number of messages added to the number of messages removed from the Timed Delivery Medium Queue per second. |
| N | TDMedQ Avg. Queue Response Time [ms] | Average time in milliseconds a message waits in the Timed Delivery Medium Queue. |
| N | TDMedQ 90% Queue Response Time [ms] | The response time in milliseconds that 90% of all messages passing through the Timed Delivery Medium Queue experience. |
| N | TDLowQ Current Depth | Current number of messages in the Timed Delivery Low Queue. |
| N | TDLowQ Now Messages In/sec | Total number of messages added to the Timed Delivery Low Queue per second. |
| N | TDLowQ Now Messages Out/sec | Total number of messages removed from the Timed Delivery Low Queue per second. |
| N | TDLowQ Now Bytes In/sec | Total number of bytes added for all messages to the Timed Delivery Low Queue per second. |
| N | TDLowQ Now Bytes Out/sec | Total number of bytes removed for all the messages from the Timed Delivery Low Queue per second. |
| N | TDLowQ Now Traffic Intensity | Ratio of the number of messages added to the number of messages removed from the Timed Delivery Low Queue per second. |
| N | TDLowQ Avg. Queue Response Time [ms] | Average time in milliseconds a message waits in the Timed Delivery Low Queue. |
| N | TDLowQ 90% Queue Response Time [ms] | The response time in milliseconds that 90% of all messages passing through the Timed Delivery Low Queue experience. |
| N | Output Waits | Total number of times output from critical client (Route or OPC) waited for ACK from MDS peer. |
| N | Average Output Wait Time | Average number of milliseconds MDS output waits to receive an ACK message from MDS peer. |
| N | Private Net Min RTT | Minimum time it took MDS to send a message over the private network and receive an ACK response from MDS peer. |
| N | Private Net Avg RTT | Average time it took MDS to send a message over the private network and receive an ACK response from MDS peer. |
| N | Private Net Max RTT | Maximum time it took MDS to send a message over the private network and receive an ACK response from MDS peer. |

### Enable optional counters

To enable Windows PerfMon counter reporting for the Message Delivery Service, you must add a new registry value (EnablePerformanceMonitor) to enable MDS process and MDS client counters.

### For the MDS process, the value is created under the MDS Process key

**Key**

    `HKEY_LOCAL_MACHINE_SOFTWARE\Cisco Systems,`
    `Inc.\ICM\`*`<Instance>`*`\`*`<node>`*`\MDS\CurrentVersion\Process`

**Name**

    EnablePerformanceMonitor

**Type**

    REG_DWORD

**Default**

    0 (disabled)

**Enabled**

    1

### For MDS clients, the value is created under each client key

**Key**

    `HKEY_LOCAL_MACHINE_SOFTWARE\Cisco Systems,`
    `Inc.\ICM\`*`<Instance>`*`\`*`<node>`*`\MDS\CurrentVersion\Clients\`*`<client>`*

**Name**

    EnablePerformanceMonitor

**Type**

    REG_DWORD

**Default**

    0 (disabled)

**Enabled**

    1

**Note** A change in this registry key is immediately detected. Performance monitor counters become enabled or disabled within 10 seconds. When Performance Monitor reporting is enabled for the MDS process, no statistical metering is reported to the MDS process log file due to overlapping functionality. When PerfMon reporting is disabled, statistical metering reporting resumes.

# QoS

**Performance Object**
    Cisco ICM QoS
**Counter Instance**
    "{Instance Name}"

**Table 23: Cisco ICM QoS**

| Always ON? | Counter Name | Description |
|---|---|---|
| N | High BytesSent/sec | High BytesSent/sec is the number of bytes per second sent to the other side over high priority connection. |

| Always ON? | Counter Name | Description |
|---|---|---|
| N | High MsgsSent/sec | High MsgsSent/sec is the number of messages sent to the other side over high priority connection. |
| N | High BytesRcvd/sec | High BytesRcvd/sec is the number of bytes received from the other side over high priority connection. |
| N | High MsgsRcvd/sec | High MsgsRcvd/sec is the number of messages received from the other side over high priority connection. |
| N | High LocalRttMean | High LocalRttMean is the mean Round Trip Time in milliseconds of high priority messages as measured by local node. |
| N | High LocalRttStdDev | High LocalRttStdDev is the standard deviation of Round Trip Time of high priority messages as measured by local node. |
| N | High RemoteRttMean | High RemoteRttMean is the mean Round Trip Time in milliseconds of high priority messages as measured by remote node. |
| N | High RemoteRttStdDev | High RemoteRttStdDev is the standard deviation of Round Trip Time of high priority messages as measured by remote node. |
| N | High Xmit NowQueueDepth | High Xmit NowQueueDepth is the current number of messages in the transmit queue for high priority traffic. |
| N | High Xmit MaxQueueDepth | High Xmit MaxQueueDepth is the maximum number of message observed in the transmit queue for high priority traffic. |
| N | High Xmit NowBytesQueued | High Xmit NowBytesQueued is the current number of bytes in the retransmit queue for high priority traffic. |
| N | High Xmit MaxBytesQueued | High Xmit MaxBytesQueued is the maximum number of bytes observed in the retransmit queue for high priority traffic. |
| N | High TotalQoSReallocations | High TotalQoSReallocations is the total number of times QoS resources had to be reallocated for high priority connection because usage has exceeded previous allocation over defined threshold levels. |
| N | Med BytesSent/sec | Med BytesSent/sec is the number of bytes per second sent to the other side over medium priority connection. |
| N | Med MsgsSent/sec | Med MsgsSent/sec is the number of messages sent to the other side over medium priority connection. |
| N | Med BytesRcvd/sec | Med BytesRcvd/sec is the number of bytes received from the other side over medium priority connection. |
| N | Med MsgsRcvd/sec | Med MsgsRcvd/sec is the number of messages received from the other side over medium priority connection. |
| N | Med LocalRttMean | Med LocalRttMean is the mean Round Trip Time in milliseconds of medium priority messages as measured by local node. |
| N | Med LocalRttStdDev | Med LocalRttStdDev is the standard deviation of Round Trip Time of medium priority messages as measured by local node. |
| N | Med RemoteRttMean | Med RemoteRttMean is the mean Round Trip Time in milliseconds of medium priority messages as measured by remote node. |

| Always ON? | Counter Name | Description |
|---|---|---|
| N | Med RemoteRttStdDev | Med RemoteRttStdDev is the standard deviation of Round Trip Time of medium priority messages as measured by remote node. |
| N | Med Xmit NowQueueDepth | Med Xmit NowQueueDepth is the current number of messages in the transmit queue for medium priority traffic. |
| N | Med Xmit MaxQueueDepth | Med Xmit MaxQueueDepth is the maximum number of message observed in the transmit queue for medium priority traffic. |
| N | Med Xmit NowBytesQueued | Med Xmit NowBytesQueued is the current number of bytes in the retransmit queue for medium priority traffic. |
| N | Med Xmit MaxBytesQueued | Med Xmit MaxBytesQueued is the maximum number of bytes observed in the retransmit queue for medium priority traffic. |
| N | Med TotalQoSReallocations | Med TotalQoSReallocations is the total number of times QoS resources had to be reallocated for medium priority connection because usage has exceeded previous allocation over defined threshold levels. |
| N | Low BytesSent/sec | Low BytesSent/sec is the number of bytes per second sent to the other side over low priority connection. |
| N | Low MsgsSent/sec | Low MsgsSent/sec is the number of messages sent to the other side over low priority connection. |
| N | Low BytesRcvd/sec | Low BytesRcvd/sec is the number of bytes received from the other side over low priority connection. |
| N | Low MsgsRcvd/sec | Low MsgsRcvd/sec is the number of messages received from the other side over low priority connection. |
| N | Low LocalRttMean | Low LocalRttMean is the mean Round Trip Time in milliseconds of low priority messages as measured by local node. |
| N | Low LocalRttStdDev | Low LocalRttStdDev is the standard deviation of Round Trip Time of low priority messages as measured by local node. |
| N | Low RemoteRttMean | Low RemoteRttMean is the mean Round Trip Time in milliseconds of low priority messages as measured by remote node. |
| N | Low RemoteRttStdDev | Low RemoteRttStdDev is the standard deviation of Round Trip Time of low priority messages as measured by remote node. |
| N | Low Xmit NowQueueDepth | Low Xmit NowQueueDepth is the current number of messages in the transmit queue for low priority traffic. |
| N | Low Xmit MaxQueueDepth | Low Xmit MaxQueueDepth is the maximum number of message observed in the transmit queue for low priority traffic. |
| N | Low Xmit NowBytesQueued | Low Xmit NowBytesQueued is the current number of bytes in the retransmit queue for low priority traffic. |
| N | Low Xmit MaxBytesQueued | Low Xmit MaxBytesQueued is the maximum number of bytes observed in the retransmit queue for low priority traffic. |
| N | Low TotalQoSReallocations | Low TotalQoSReallocations is the total number of times QoS resources had to be reallocated for low priority connection because usage has exceeded previous allocation over defined threshold levels. |

## Enable Optional Counters

Because there is overhead in maintaining QoS Performance Monitoring counters, the performance monitoring feature is turned off by default. To enable this feature, change the following registry key value to 1and cycle the application process.

**Key**

```
HKEY_LOCAL_MACHINE_SOFTWARE\Cisco Systems,
Inc.\ICM\<Instance>\<node>\DMP\CurrentVersion
```

**Name**

EnablePerformanceMonitor

**Type**

REG_DWORD

**Default**

0 (disabled)

**Enable**

1

**Note** The amount of overhead is dependent on the periodic update interval. This interval should be set reasonably high to minimize the impact on the system.