

[.NET CLR Data](#)
[.NET CLR Exceptions](#)
[.NET CLR Interop](#)
[.NET CLR Jit](#)
[.NET CLR Loading](#)
[.NET CLR LocksAndThreads](#)
[.NET CLR Memory](#)
[.NET CLR Networking](#)
[.NET CLR Networking 4.0.0.0](#)
[.NET CLR Remoting](#)
[.NET CLR Security](#)
[.NET Data Provider for Oracle](#)
[.NET Data Provider for SqlServer](#)
[.NET Memory Cache 4.0](#)
[ASP.NET](#)
[ASP.NET Applications](#)
[ASP.NET Apps v2.0.50727](#)
[ASP.NET Apps v4.0.30319](#)
[ASP.NET State Service](#)
[ASP.NET v2.0.50727](#)
[ASP.NET v4.0.30319](#)
[Browser](#)
[Cache](#)
[Distributed Transaction Coordinator](#)
[ICMP](#)
[IP](#)
[Job Object](#)
[Job Object Details](#)
[LogicalDisk](#)
[Memory](#)
[MSDTC Bridge 3.0.0.0](#)
[MSDTC Bridge 4.0.0.0](#)
[NBT Connection](#)
[Network Interface](#)
[Objects](#)
[Paging File](#)
[PhysicalDisk](#)
[Print Queue](#)
[Process](#)
[Processor](#)
[PSched Flow](#)
[PSched Pipe](#)
[Redirector](#)
[RSVP Interfaces](#)
[RSVP Service](#)
[Server](#)
[Server Work Queues](#)
[ServiceModelEndpoint 3.0.0.0](#)
[ServiceModelEndpoint 4.0.0.0](#)
[ServiceModelOperation 3.0.0.0](#)
[ServiceModelOperation 4.0.0.0](#)
[ServiceModelService 3.0.0.0](#)
[ServiceModelService 4.0.0.0](#)
[SMSvcHost 3.0.0.0](#)
[SMSvcHost 4.0.0.0](#)
[System](#)

[TCP](#)
[Telephony](#)
[Terminal Services](#)
[Terminal Services Session](#)
[Thread](#)
[UDP](#)
[WF \(System.Workflow\) 4.0.0.0](#)
[Windows Workflow Foundation](#)
[WMI Objects](#)

List of all performance categories, their instances (if applicable) and their performance counters.

Computername: WVXP3073

Date and time: 2012-10-30 14:16:32

Category: .NET CLR Data

CategoryType: MultiInstance.

.Net CLR Data

Category: .NET CLR Exceptions

CategoryType: MultiInstance.

Runtime statistics on CLR exception handling.

Instance: powershell

Counter Name	Counter Type	Counter Description
# of Exceps Thrown	NumberOfItems32	This counter displays the total number of exceptions thrown since the start of the application. These include both .NET exceptions and unmanaged exceptions that get converted into .NET exceptions e.g. null pointer reference exception in unmanaged code would get re-thrown in managed code as a .NET System.NullReferenceException; this counter includes both handled and unhandled exceptions. Exceptions that are re-thrown would get counted again. Exceptions should only occur in rare situations and not in the normal control flow of the program.
# of Exceps Thrown / sec	RateOfCountsPerSecond32	This counter displays the number of exceptions thrown per second. These include both .NET exceptions and unmanaged exceptions that get converted into .NET exceptions e.g. null pointer reference exception in unmanaged code would get re-thrown in managed code as a .NET System.NullReferenceException; this counter includes both handled and unhandled exceptions. Exceptions should only occur

Counter Name	Counter Type	Counter Description
		in rare situations and not in the normal control flow of the program; this counter was designed as an indicator of potential performance problems due to large (>100s) rate of exceptions thrown. This counter is not an average over time; it displays the difference between the values observed in the last two samples divided by the duration of the sample interval.
# of Filters / sec	RateOfCountsPerSecond32	This counter displays the number of .NET exception filters executed per second. An exception filter evaluates whether an exception should be handled or not. This counter tracks the rate of exception filters evaluated; irrespective of whether the exception was handled or not. This counter is not an average over time; it displays the difference between the values observed in the last two samples divided by the duration of the sample interval.
# of Finallys / sec	RateOfCountsPerSecond32	This counter displays the number of finally blocks executed per second. A finally block is guaranteed to be executed regardless of how the try block was exited. Only the finally blocks that are executed for an exception are counted; finally blocks on normal code paths are not counted by this counter. This counter is not an average over time; it displays the difference between the values observed in the last two samples divided by the duration of the sample interval.
Throw To Catch Depth / sec	RateOfCountsPerSecond32	This counter displays the number of stack frames traversed from the frame that threw the .NET exception to the frame that handled the exception per second. This counter resets to 0 when an exception handler is entered; so nested exceptions would show the handler to handler stack depth. This counter is not an average over time; it displays the difference between the values observed in the last two samples divided by the duration of the sample interval.

Instance: powershell_ise

Instance: _Global_

Instance: PresentationFontCache

Category: .NET CLR Interop

CategoryType: MultiInstance.

Stats for CLR interop.

Instance: powershell

Counter Name	Counter Type	Counter Description
# of CCWs	NumberOfItems32	This counter displays the current number of Com-Callable-Wrappers (CCWs). A CCW is a proxy for the .NET managed object being referenced from unmanaged COM client(s). This counter was designed to indicate the number of managed objects being referenced by unmanaged COM code.
# of Stubs	NumberOfItems32	This counter displays the current number of stubs created by the CLR. Stubs are responsible for marshalling arguments and return values from managed to unmanaged code and vice versa; during a COM Interop call or PInvoke call.
# of marshalling	NumberOfItems32	This counter displays the total number of times arguments and return values have been marshaled from managed to unmanaged code and vice versa since the start of the application. This counter is not incremented if the stubs are inlined. (Stubs are responsible for marshalling arguments and return values). Stubs usually get inlined if the marshalling overhead is small.
# of TLB imports / sec	NumberOfItems32	Reserved for future use.
# of TLB exports / sec	NumberOfItems32	Reserved for future use.

Instance: powershell_ise

Instance: _Global_

Instance: PresentationFontCache

Category: .NET CLR Jit

CategoryType: MultiInstance.

Stats for CLR Jit.

Instance: powershell

Counter Name	Counter Type	Counter Description
# of	NumberOfItems32	This counter displays the total number of methods compiled

Counter Name	Counter Type	Counter Description
Methods Jitted		Just-In-Time (JIT) by the CLR JIT compiler since the start of the application. This counter does not include the pre-jitted methods.
# of IL Bytes Jitted	NumberOfItems32	This counter displays the total IL bytes jitted since the start of the application. This counter is exactly equivalent to the "Total # of IL Bytes Jitted" counter.
Total # of IL Bytes Jitted	NumberOfItems32	This counter displays the total IL bytes jitted since the start of the application. This counter is exactly equivalent to the "# of IL Bytes Jitted" counter.
IL Bytes Jitted / sec	RateOfCountsPerSecond32	This counter displays the rate at which IL bytes are jitted per second. This counter is not an average over time; it displays the difference between the values observed in the last two samples divided by the duration of the sample interval.
Standard Jit Failures	NumberOfItems32	This counter displays the peak number of methods the JIT compiler has failed to JIT since the start of the application. This failure can occur if the IL cannot be verified or if there was an internal error in the JIT compiler.
% Time in Jit	RawFraction	This counter displays the percentage of elapsed time spent in JIT compilation since the last JIT compilation phase. This counter is updated at the end of every JIT compilation phase. A JIT compilation phase is the phase when a method and its dependencies are being compiled.
Not Displayed	RawBase	Not Displayed.

Instance: powershell_ise

Instance: _Global_

Instance: PresentationFontCache

Category: .NET CLR Loading

CategoryType: MultiInstance.

Statistics for CLR Class Loader.

Instance: powershell

Counter Name	Counter Type	Counter Description
Current Classes Loaded	NumberOfItems32	This counter displays the current number of classes loaded in all Assemblies.
Total Classes Loaded	NumberOfItems32	This counter displays the cumulative number of classes loaded in all Assemblies since the start of this application.
Rate of Classes Loaded	RateOfCountsPerSecond32	This counter displays the number of classes loaded per second in all Assemblies. This counter is not an average over time; it displays the difference between the values observed in the last two samples divided by the duration of the sample interval.
Current appdomains	NumberOfItems32	This counter displays the current number of AppDomains loaded in this application. AppDomains (application domains) provide a secure and versatile unit of processing that the CLR can use to provide isolation between applications running in the same process.
Total Appdomains	NumberOfItems32	This counter displays the peak number of AppDomains loaded since the start of this application. AppDomains (application domains) provide a secure and versatile unit of processing that the CLR can use to provide isolation between applications running in the same process.
Rate of appdomains	RateOfCountsPerSecond32	This counter displays the number of AppDomains loaded per second. AppDomains (application domains) provide a secure and versatile unit of processing that the CLR can use to provide isolation between applications running in the same process. This counter is not an average over time; it displays the difference between the values observed in the last two samples divided by the duration of the sample interval.
Current Assemblies	NumberOfItems32	This counter displays the current number of Assemblies loaded across all AppDomains in this application. If the Assembly is loaded as domain-neutral from multiple AppDomains then this counter is incremented once only. Assemblies can be loaded as domain-neutral when their code can be shared by all AppDomains or they can be loaded as domain-specific when their code is private to the AppDomain.
Total Assemblies	NumberOfItems32	This counter displays the total number of Assemblies loaded since the start of this application. If the Assembly is loaded as domain-neutral from multiple AppDomains then this counter is incremented once only. Assemblies can be loaded as domain-neutral when their code can be shared by all

Counter Name	Counter Type	Counter Description
		AppDomains or they can be loaded as domain-specific when their code is private to the AppDomain.
Rate of Assemblies	RateOfCountsPerSecond32	This counter displays the number of Assemblies loaded across all AppDomains per second. If the Assembly is loaded as domain-neutral from multiple AppDomains then this counter is incremented once only. Assemblies can be loaded as domain-neutral when their code can be shared by all AppDomains or they can be loaded as domain-specific when their code is private to the AppDomain. This counter is not an average over time; it displays the difference between the values observed in the last two samples divided by the duration of the sample interval.
% Time Loading	CounterTimer	Reserved for future use.
Assembly Search Length	NumberOfItems32	Reserved for future use.
Total # of Load Failures	NumberOfItems32	This counter displays the peak number of classes that have failed to load since the start of the application. These load failures could be due to many reasons like inadequate security or illegal format. Full details can be found in the profiling services help.
Rate of Load Failures	RateOfCountsPerSecond32	This counter displays the number of classes that failed to load per second. This counter is not an average over time; it displays the difference between the values observed in the last two samples divided by the duration of the sample interval. These load failures could be due to many reasons like inadequate security or illegal format. Full details can be found in the profiling services help.
Bytes in Loader Heap	NumberOfItems32	This counter displays the current size (in bytes) of the memory committed by the class loader across all AppDomains. (Committed memory is the physical memory for which space has been reserved on the disk paging file.)
Total appdomains unloaded	NumberOfItems32	This counter displays the total number of AppDomains unloaded since the start of the application. If an AppDomain is loaded and unloaded multiple times this counter would count each of those unloads as separate.
Rate of appdomains unloaded	RateOfCountsPerSecond32	This counter displays the number of AppDomains unloaded per second. This counter is not an average over time; it displays the difference between the values observed in the last two samples divided by the duration of the sample

Counter Name	Counter Type	Counter Description
		interval.

Instance: powershell_ise

Instance: _Global_

Instance: PresentationFontCache

Category: .NET CLR LocksAndThreads

CategoryType: MultiInstance.

Stats for CLR Locks and Threads.

Instance: powershell

Counter Name	Counter Type	Counter Description
Total # of Contentions	NumberOfItems32	This counter displays the total number of times threads in the CLR have attempted to acquire a managed lock unsuccessfully. Managed locks can be acquired in many ways; by the "lock" statement in C# or by calling System.Monitor.Enter or by using MethodImplOptions.Synchronized custom attribute.
Contention Rate / sec	RateOfCountsPerSecond32	Rate at which threads in the runtime attempt to acquire a managed lock unsuccessfully. Managed locks can be acquired in many ways; by the "lock" statement in C# or by calling System.Monitor.Enter or by using MethodImplOptions.Synchronized custom attribute.
Current Queue Length	NumberOfItems32	This counter displays the total number of threads currently waiting to acquire some managed lock in the application. This counter is not an average over time; it displays the last observed value.
Queue Length Peak	NumberOfItems32	This counter displays the total number of threads that waited to acquire some managed lock since the start of the application.
Queue Length / sec	RateOfCountsPerSecond32	This counter displays the number of threads per second waiting to acquire some lock in the application. This counter is not an average over time; it displays the difference between the values observed in the last two samples divided by the

Counter Name	Counter Type	Counter Description
		duration of the sample interval.
# of current logical Threads	NumberOfItems32	This counter displays the number of current .NET thread objects in the application. A .NET thread object is created either by new System.Threading.Thread or when an unmanaged thread enters the managed environment. This counter maintains the count of both running and stopped threads. This counter is not an average over time; it just displays the last observed value.
# of current physical Threads	NumberOfItems32	This counter displays the number of native OS threads created and owned by the CLR to act as underlying threads for .NET thread objects. This counter's value does not include the threads used by the CLR in its internal operations; it is a subset of the threads in the OS process.
# of current recognized threads	NumberOfItems32	This counter displays the number of threads that are currently recognized by the CLR; they have a corresponding .NET thread object associated with them. These threads are not created by the CLR; they are created outside the CLR but have since run inside the CLR at least once. Only unique threads are tracked; threads with same thread ID re-entering the CLR or recreated after thread exit are not counted twice.
# of total recognized threads	NumberOfItems32	This counter displays the total number of threads that have been recognized by the CLR since the start of this application; these threads have a corresponding .NET thread object associated with them. These threads are not created by the CLR; they are created outside the CLR but have since run inside the CLR at least once. Only unique threads are tracked; threads with same thread ID re-entering the CLR or recreated after thread exit are not counted twice.
rate of recognized threads / sec	RateOfCountsPerSecond32	This counter displays the number of threads per second that have been recognized by the CLR; these threads have a corresponding .NET thread object associated with them. These threads are not created by the CLR; they are created outside the CLR but have since run inside the CLR at least once. Only unique threads are tracked; threads with same thread ID re-entering the CLR or recreated after thread exit are not counted twice. This counter is not an average over time; it displays the difference between the values observed in the last two samples divided by the duration of the sample interval.

Instance: powershell_ise

Instance: _Global_

Instance: PresentationFontCache

Category: .NET CLR Memory

CategoryType: MultiInstance.

Counters for CLR Garbage Collected heap.

Instance: powershell

Counter Name	Counter Type	Counter Description
# Gen 0 Collections	NumberOfItems32	This counter displays the number of times the generation 0 objects (youngest; most recently allocated) are garbage collected (Gen 0 GC) since the start of the application. Gen 0 GC occurs when the available memory in generation 0 is not sufficient to satisfy an allocation request. This counter is incremented at the end of a Gen 0 GC. Higher generation GCs include all lower generation GCs. This counter is explicitly incremented when a higher generation (Gen 1 or Gen 2) GC occurs. <u>_Global_</u> counter value is not accurate and should be ignored. This counter displays the last observed value.
# Gen 1 Collections	NumberOfItems32	This counter displays the number of times the generation 1 objects are garbage collected since the start of the application. The counter is incremented at the end of a Gen 1 GC. Higher generation GCs include all lower generation GCs. This counter is explicitly incremented when a higher generation (Gen 2) GC occurs. <u>_Global_</u> counter value is not accurate and should be ignored. This counter displays the last observed value.
# Gen 2 Collections	NumberOfItems32	This counter displays the number of times the generation 2 objects (older) are garbage collected since the start of the application. The counter is incremented at the end of a Gen 2 GC (also called full GC). <u>_Global_</u> counter value is not accurate and should be ignored. This counter displays the last observed value.
Promoted Memory from Gen 0	NumberOfItems32	This counter displays the bytes of memory that survive garbage collection (GC) and are promoted from generation 0 to generation 1; objects that are promoted just because they are waiting to be finalized are not included in this counter. This counter displays the value observed at the end of the last GC; its not a cumulative counter.
Promoted Memory from Gen 1	NumberOfItems32	This counter displays the bytes of memory that survive garbage collection (GC) and are promoted

Counter Name	Counter Type	Counter Description
		from generation 1 to generation 2; objects that are promoted just because they are waiting to be finalized are not included in this counter. This counter displays the value observed at the end of the last GC; its not a cumulative counter. This counter is reset to 0 if the last GC was a Gen 0 GC only.
Gen 0 Promoted Bytes/Sec	RateOfCountsPerSecond32	This counter displays the bytes per second that are promoted from generation 0 (youngest) to generation 1; objects that are promoted just because they are waiting to be finalized are not included in this counter. Memory is promoted when it survives a garbage collection. This counter was designed as an indicator of relatively long-lived objects being created per sec. This counter displays the difference between the values observed in the last two samples divided by the duration of the sample interval.
Gen 1 Promoted Bytes/Sec	RateOfCountsPerSecond32	This counter displays the bytes per second that are promoted from generation 1 to generation 2 (oldest); objects that are promoted just because they are waiting to be finalized are not included in this counter. Memory is promoted when it survives a garbage collection. Nothing is promoted from generation 2 since it is the oldest. This counter was designed as an indicator of very long-lived objects being created per sec. This counter displays the difference between the values observed in the last two samples divided by the duration of the sample interval.
Promoted Finalization-Memory from Gen 0	NumberOfItems32	This counter displays the bytes of memory that are promoted from generation 0 to generation 1 just because they are waiting to be finalized. This counter displays the value observed at the end of the last GC; its not a cumulative counter.
Process ID	NumberOfItems32	This counter displays the process ID of the CLR process instance being monitored.
Gen 0 heap size	NumberOfItems32	This counter displays the maximum bytes that can be allocated in generation 0 (Gen 0); its does not indicate the current number of bytes allocated in Gen 0. A Gen 0 GC is triggered when the allocations since the last GC exceed this size. The Gen 0 size is tuned by the Garbage Collector and can change during the execution of the application. At the end of a Gen 0 collection the size of the Gen 0 heap is infact 0 bytes; this counter displays the size (in bytes) of allocations that would trigger the next Gen 0 GC. This counter is updated at the end of a GC; its not updated on every

Counter Name	Counter Type	Counter Description
		allocation.
Gen 1 heap size	NumberOfItems32	This counter displays the current number of bytes in generation 1 (Gen 1); this counter does not display the maximum size of Gen 1. Objects are not directly allocated in this generation; they are promoted from previous Gen 0 GCs. This counter is updated at the end of a GC; its not updated on every allocation.
Gen 2 heap size	NumberOfItems32	This counter displays the current number of bytes in generation 2 (Gen 2). Objects are not directly allocated in this generation; they are promoted from Gen 1 during previous Gen 1 GCs. This counter is updated at the end of a GC; its not updated on every allocation.
Large Object Heap size	NumberOfItems32	This counter displays the current size of the Large Object Heap in bytes. Objects greater than 20 KBytes are treated as large objects by the Garbage Collector and are directly allocated in a special heap; they are not promoted through the generations. This counter is updated at the end of a GC; its not updated on every allocation.
Finalization Survivors	NumberOfItems32	This counter displays the number of garbage collected objects that survive a collection because they are waiting to be finalized. If these objects hold references to other objects then those objects also survive but are not counted by this counter; the "Promoted Finalization-Memory from Gen 0" and "Promoted Finalization-Memory from Gen 1" counters represent all the memory that survived due to finalization. This counter is not a cumulative counter; its updated at the end of every GC with count of the survivors during that particular GC only. This counter was designed to indicate the extra overhead that the application might incur because of finalization.
# GC Handles	NumberOfItems32	This counter displays the current number of GC Handles in use. GCHandles are handles to resources external to the CLR and the managed environment. Handles occupy small amounts of memory in the GCHeap but potentially expensive unmanaged resources.
Allocated Bytes/sec	RateOfCountsPerSecond32	This counter displays the rate of bytes per second allocated on the GC Heap. This counter is updated at the end of every GC; not at each allocation. This counter is not an average over time; it displays the difference between the values observed in the last two samples divided by the duration of the sample

Counter Name	Counter Type	Counter Description
		interval.
# Induced GC	NumberOfItems32	This counter displays the peak number of times a garbage collection was performed because of an explicit call to GC.Collect. Its a good practice to let the GC tune the frequency of its collections.
% Time in GC	RawFraction	% Time in GC is the percentage of elapsed time that was spent in performing a garbage collection (GC) since the last GC cycle. This counter is usually an indicator of the work done by the Garbage Collector on behalf of the application to collect and compact memory. This counter is updated only at the end of every GC and the counter value reflects the last observed value; its not an average.
Not Displayed	RawBase	Not Displayed.
# Bytes in all Heaps	NumberOfItems32	This counter is the sum of four other counters; Gen 0 Heap Size; Gen 1 Heap Size; Gen 2 Heap Size and the Large Object Heap Size. This counter indicates the current memory allocated in bytes on the GC Heaps.
# Total committed Bytes	NumberOfItems32	This counter displays the amount of virtual memory (in bytes) currently committed by the Garbage Collector. (Committed memory is the physical memory for which space has been reserved on the disk paging file).
# Total reserved Bytes	NumberOfItems32	This counter displays the amount of virtual memory (in bytes) currently reserved by the Garbage Collector. (Reserved memory is the virtual memory space reserved for the application but no disk or main memory pages have been used.)
# of Pinned Objects	NumberOfItems32	This counter displays the number of pinned objects encountered in the last GC. This counter tracks the pinned objects only in the heaps that were garbage collected e.g. a Gen 0 GC would cause enumeration of pinned objects in the generation 0 heap only. A pinned object is one that the Garbage Collector cannot move in memory.
# of Sink Blocks in use	NumberOfItems32	This counter displays the current number of sync blocks in use. Sync blocks are per-object data structures allocated for storing synchronization information. Sync blocks hold weak references to managed objects and need to be scanned by the Garbage Collector. Sync blocks are not limited to storing synchronization information and can also store

Counter Name	Counter Type	Counter Description
		COM interop metadata. This counter was designed to indicate performance problems with heavy use of synchronization primitives.

Instance: powershell_ise

Instance: _Global_

Instance: PresentationFontCache

Category: .NET CLR Networking

CategoryType: MultiInstance.

Help not available.

Category: .NET CLR Networking 4.0.0.0

CategoryType: MultiInstance.

Counters for classes in the System.Net namespace.

Category: .NET CLR Remoting

CategoryType: MultiInstance.

Stats for CLR Remoting.

Instance: powershell

Counter Name	Counter Type	Counter Description
Remote Calls/sec	RateOfCountsPerSecond32	This counter displays the number of remote procedure calls invoked per second. A remote procedure call is a call on any object outside the caller's AppDomain. This counter is not an average over time; it displays the difference between the values observed in the last two samples divided by the duration of the sample interval.
Total Remote Calls	NumberOfItems32	This counter displays the total number of remote procedure calls invoked since the start of this application. A remote procedure call is a call on any object outside the caller's AppDomain.

Counter Name	Counter Type	Counter Description
Channels	NumberOfItems32	This counter displays the total number of remoting channels registered across all AppDomains since the start of the application. Channels are used to transport messages to and from remote objects.
Context Proxies	NumberOfItems32	This counter displays the total number of remoting proxy objects created in this process since the start of the process. Proxy object acts as a representative of the remote objects and ensures that all calls made on the proxy are forwarded to the correct remote object instance.
Context-Bound Classes Loaded	NumberOfItems32	This counter displays the current number of context-bound classes loaded. Classes that can be bound to a context are called context-bound classes; context-bound classes are marked with Context Attributes which provide usage rules for synchronization; thread affinity; transactions etc.
Context-Bound Objects Alloc / sec	RateOfCountsPerSecond32	This counter displays the number of context-bound objects allocated per second. Instances of classes that can be bound to a context are called context-bound objects; context-bound classes are marked with Context Attributes which provide usage rules for synchronization; thread affinity; transactions etc. This counter is not an average over time; it displays the difference between the values observed in the last two samples divided by the duration of the sample interval.
Contexts	NumberOfItems32	This counter displays the current number of remoting contexts in the application. A context is a boundary containing a collection of objects with the same usage rules like synchronization; thread affinity; transactions etc.

Instance: powershell_ise

Instance: _Global_

Instance: PresentationFontCache

Category: .NET CLR Security

CategoryType: MultiInstance.

Stats for CLR Security.

Instance: powershell

Counter Name	Counter Type	Counter Description
Total Runtime Checks	NumberOfItems32	This counter displays the total number of runtime Code Access Security (CAS) checks performed since the start of the application. Runtime CAS checks are performed when a caller makes a call to a callee demanding a particular permission; the runtime check is made on every call by the caller; the check is done by examining the current thread stack of the caller. This counter used together with "Stack Walk Depth" is indicative of performance penalty for security checks.
% Time Sig. Authenticating	CounterTimer	Reserved for future use.
# Link Time Checks	NumberOfItems32	This counter displays the total number of linktime Code Access Security (CAS) checks since the start of the application. Linktime CAS checks are performed when a caller makes a call to a callee demanding a particular permission at JIT compile time; linktime check is performed once per caller. This count is not indicative of serious performance issues; its indicative of the security system activity.
% Time in RT checks	RawFraction	This counter displays the percentage of elapsed time spent in performing runtime Code Access Security (CAS) checks since the last such check. CAS allows code to be trusted to varying degrees and enforces these varying levels of trust depending on code identity. This counter is updated at the end of a runtime security check; it represents the last observed value; its not an average.
Not Displayed	RawBase	Not Displayed.
Stack Walk Depth	NumberOfItems32	This counter displays the depth of the stack during that last runtime Code Access Security check. Runtime Code Access Security check is performed by crawling the stack. This counter is not an average; it just displays the last observed value.

Instance: powershell_ise

Instance: _Global_

Instance: PresentationFontCache

Category: .NET Data Provider for Oracle

CategoryType: MultiInstance.

Counters for System.Data.OracleClient

Category: .NET Data Provider for SqlServer

CategoryType: MultiInstance.

Counters for System.Data.SqlClient

Category: .NET Memory Cache 4.0

CategoryType: MultiInstance.

System.Runtime.Caching.MemoryCache Performance Counters

Category: ASP.NET

CategoryType: SingleInstance.

ASP.NET global performance counters

Counter Name	Counter Type	Counter Description
State Server Sessions Total	NumberOfItems3 2	The number of sessions total.
Requests Current	NumberOfItems3 2	The current number of requests, including those that are queued, currently executing, or waiting to be written to the client. Under the ASP.NET process model, when this counter exceeds the requestQueueLimit defined in the processModel configuration section, ASP.NET will begin rejecting requests.
State Server Sessions Abandoned	NumberOfItems3 2	The number of sessions that have been explicitly abandoned.
State Server Sessions Timed Out	NumberOfItems3 2	The number of sessions timed out.
Audit Success Events Raised	NumberOfItems3 2	Number of audit successes in the application since it was started.
Request Error Events Raised	NumberOfItems3 2	Number of runtime error events raised since the application was started.
Infrastructure Error Events Raised	NumberOfItems3 2	Number of HTTP error events raised since the application was started.

Counter Name	Counter Type	Counter Description
Audit Failure Events Raised	NumberOfItems32	Number of audit failures in the application since it was started.
Error Events Raised	NumberOfItems32	Number of error events raised since the application was started.
State Server Sessions Active	NumberOfItems32	The current number of sessions currently active.
Requests Disconnected	NumberOfItems32	The number of requests disconnected due to communication errors or user terminated.
Request Execution Time	NumberOfItems32	The number of milliseconds that it took to execute the most recent request.
Application Restarts	NumberOfItems32	Number of times the application has been restarted during the web server's lifetime.
Applications Running	NumberOfItems32	Number of currently running web applications.
Requests Rejected	NumberOfItems32	The number of requests rejected because the request queue was full.
Worker Process Restarts	NumberOfItems32	Number of times a worker process has restarted on the machine.
Request Wait Time	NumberOfItems32	The number of milliseconds the most recent request was waiting in the queue.
Requests Queued	NumberOfItems32	The number of requests waiting to be processed.
Worker Processes Running	NumberOfItems32	Number of worker processes running on the machine.

Category: ASP.NET Applications

CategoryType: MultiInstance.

ASP.NET application performance counters

Instance: __Total__

Counter Name	Counter Type	Counter Description
Anonymous Requests	NumberOfItems32	Number of requests utilizing anonymous authentication.
Anonymous Requests/Sec	RateOfCountsPerSecond32	Number of Authentication Anonymous Requests/Sec
Cache Total Entries	NumberOfItems32	Total number of entries within the cache (both internal and user added)
Cache Total Turnover Rate	RateOfCountsPerSecond32	Number of additions and removals to the total cache per second.
Cache Total Hits	NumberOfItems32	Total number of hits from the cache.
Cache Total Misses	NumberOfItems32	Total number of cache misses.
Cache Total Hit Ratio	RawFraction	Ratio of hits from all cache calls.
Cache Total Hit Ratio Base	RawBase	Cache Total Hit Ratio Base
Cache API Entries	NumberOfItems32	Total number of entries within the cache added by the user.
Cache API Turnover Rate	RateOfCountsPerSecond32	Number of additions and removals to the API cache per second.
Cache API Hits	NumberOfItems32	Number of cache hits from user code.
Cache API Misses	NumberOfItems32	Number of cache misses called from user code.
Cache API Hit Ratio	RawFraction	Ratio of hits called from user code.
Cache API Hit Ratio Base	RawBase	Cache API Hit Ratio Base
Output Cache Entries	NumberOfItems32	Current number of entries in the output cache.
Output Cache	RateOfCountsPerSecond32	Number of additions and removals to the output cache

Counter Name	Counter Type	Counter Description
Turnover Rate	2	per second.
Output Cache Hits	NumberOfItems32	Total number of output cacheable requests served from the output cache.
Output Cache Misses	NumberOfItems32	Total number of output cacheable requests not served from the output cache.
Output Cache Hit Ratio	RawFraction	Ratio of hits to requests for output cacheable requests.
Output Cache Hit Ratio Base	RawBase	Output Cache Hit Ratio Base
Compilations Total	NumberOfItems32	Number of .aspx, .ascx, .ashx, .asmx, or .aspx source files dynamically compiled.
Debugging Requests	NumberOfItems32	Number of debugging requests processed.
Errors During Preprocessing	NumberOfItems32	Number of errors that have occurred during parsing and configuration.
Errors During Compilation	NumberOfItems32	Number of errors that have occurred during compilation.
Errors During Execution	NumberOfItems32	Number of errors that have occurred during the processing of a request.
Errors Unhandled During Execution	NumberOfItems32	Number of errors not handled by user code, but by the default error handler.
Errors Unhandled During Execution/Sec	RateOfCountsPerSecond32	Rate of unhandled errors.
Errors Total	NumberOfItems32	Total number of errors occurred.
Errors Total/Sec	RateOfCountsPerSecond32	Rate of errors occurred.
Pipeline Instance Count	NumberOfItems32	Number of active pipeline instances.

Counter Name	Counter Type	Counter Description
Request Bytes In Total	NumberOfItems32	The total size, in bytes, of all requests.
Request Bytes Out Total	NumberOfItems32	The total size, in bytes, of responses sent to a client. This does not include standard HTTP response headers.
Requests Executing	NumberOfItems32	The number of requests currently executing.
Requests Failed	NumberOfItems32	Total number of failed requests.
Requests Not Found	NumberOfItems32	The number of requests for resources that were not found.
Requests Not Authorized	NumberOfItems32	Number of requests failed due to unauthorized access.
Requests In Application Queue	NumberOfItems32	The number of requests in the application request queue.
Requests Timed Out	NumberOfItems32	The number of requests that timed out.
Requests Succeeded	NumberOfItems32	The number of requests that executed successfully.
Requests Total	NumberOfItems32	The total number of requests since the application was started.
Requests/Sec	RateOfCountsPerSecond32	The number of requests executed per second.
Sessions Active	NumberOfItems32	The current number of sessions currently active.
Sessions Abandoned	NumberOfItems32	The number of sessions that have been explicitly abandoned.
Sessions Timed Out	NumberOfItems32	The number of sessions timed out.
Sessions Total	NumberOfItems32	Total number of sessions since the application was started.

Counter Name	Counter Type	Counter Description
Transactions Aborted	NumberOfItems32	The number of transactions aborted.
Transactions Committed	NumberOfItems32	The number of transactions committed.
Transactions Pending	NumberOfItems32	Number of transactions in progress.
Transactions Total	NumberOfItems32	The total number of transactions since the application was started.
Transactions/Sec	RateOfCountsPerSecond32	Transactions started per second.
Session State Server connections total	NumberOfItems32	The total number of connections to the State Server used by session state.
Session SQL Server connections total	NumberOfItems32	The total number of connections to the SQL Server used by session state.
Events Raised	NumberOfItems32	Total number of instrumentation events raised since the application was started.
Events Raised/Sec	RateOfCountsPerSecond32	Total number of instrumentation events per second.
Application Lifetime Events	NumberOfItems32	Number of application events raised since the application was started.
Application Lifetime Events/Sec	RateOfCountsPerSecond32	Number of application events raised per second.
Error Events Raised	NumberOfItems32	Number of error events raised since the application was started.
Error Events Raised/Sec	RateOfCountsPerSecond32	Number of error events per second.
Request Error Events Raised	NumberOfItems32	Number of runtime error events raised since the application was started.
Request Error Events	RateOfCountsPerSecond32	Number of runtime error events per second.

Counter Name	Counter Type	Counter Description
Raised/Sec		
Infrastructure Error Events Raised	NumberOfItems32	Number of HTTP error events raised since the application was started.
Infrastructure Error Events Raised/Sec	RateOfCountsPerSecond32	Number of HTTP error events raised per second.
Request Events Raised	NumberOfItems32	Number of request events raised since the application was started
Request Events Raised/Sec	RateOfCountsPerSecond32	Number of request events raised per second.
Audit Success Events Raised	NumberOfItems32	Number of audit successes in the application since it was started.
Audit Failure Events Raised	NumberOfItems32	Number of audit failures in the application since it was started.
Membership Authentication Success	NumberOfItems32	Number of successful membership credential validations since the application was started.
Membership Authentication Failure	NumberOfItems32	Number of failed membership credential validations since the application was started.
Forms Authentication Success	NumberOfItems32	Number of successful forms authentication ticket validations since the application was started.
Forms Authentication Failure	NumberOfItems32	Number of failed forms authentication ticket validations since the application was started.
Viewstate MAC Validation Failure	NumberOfItems32	Number of viewstate MAC validations that failed since the application was started.
Request Execution Time	NumberOfItems32	The number of milliseconds that it took to execute the most recent request.
Requests Disconnected	NumberOfItems32	The number of requests disconnected due to

Counter Name	Counter Type	Counter Description
		communication errors or user terminated.
Requests Rejected	NumberOfItems32	The number of requests rejected because the application request queue was full.
Request Wait Time	NumberOfItems32	The number of milliseconds the most recent request was waiting in the queue.
Cache % Machine Memory Limit Used	RawFraction	The amount of physical memory used by the machine divided by the physical memory limit for the cache, as a percentage. When this reaches 100%, half of the cache entries will be forcibly removed. The __Total__ instance is the average of all instances, and therefore cannot be used to determine when cache entries will be forcibly removed.
Cache % Machine Memory Limit Used Base	RawBase	Cache % Machine Memory Limit Used Base
Cache % Process Memory Limit Used	RawFraction	The value of private bytes for the worker process divided by the private bytes memory limit for the cache, as a percentage. When this reaches 100%, half of the cache entries will be forcibly removed. The __Total__ instance is the average of all instances, and therefore cannot be used to determine when cache entries will be forcibly removed.
Cache % Process Memory Limit Used Base	RawBase	Cache % Process Memory Limit Used Base
Cache Total Trims	NumberOfItems32	Total number of entries forcibly removed from the cache due to memory pressure.
Cache API Trims	NumberOfItems32	Total number of entries forcibly removed from the cache due to memory pressure that were originally inserted into the cache using one of the public cache APIs.
Output Cache Trims	NumberOfItems32	Total number of entries forcibly removed from the cache due to memory pressure that were originally inserted into the cache by the output cache feature.
% Managed Processor Time (estimated)	RawFraction	Estimated percentage of elapsed time that the processor spends executing managed application code. This counter only tracks processor time of managed threads in the application. It does not include additional processor time spent executing on non-managed threads. Note that this counter is only updated with new data every five

Counter Name	Counter Type	Counter Description
		seconds.
% Managed Processor Time Base (estimated)	RawBase	% Managed Processor Time Base (estimated)
Managed Memory Used (estimated)	NumberOfItems32	Estimated managed heap memory consumption (in KB) by the application. The accuracy of this counter varies depending on the duration of elapsed time since the last full managed memory heap collection. Note that this counter is only updated with new data every five seconds.

Category: ASP.NET Apps v2.0.50727

CategoryType: MultiInstance.

ASP.NET application performance counters

Instance: Total

Counter Name	Counter Type	Counter Description
Anonymous Requests	NumberOfItems32	Number of requests utilizing anonymous authentication.
Anonymous Requests/Sec	RateOfCountsPerSecond32	Number of Authentication Anonymous Requests/Sec
Cache Total Entries	NumberOfItems32	Total number of entries within the cache (both internal and user added)
Cache Total Turnover Rate	RateOfCountsPerSecond32	Number of additions and removals to the total cache per second.
Cache Total Hits	NumberOfItems32	Total number of hits from the cache.
Cache Total Misses	NumberOfItems32	Total number of cache misses.
Cache Total Hit Ratio	RawFraction	Ratio of hits from all cache calls.

Counter Name	Counter Type	Counter Description
Cache Total Hit Ratio Base	RawBase	Cache Total Hit Ratio Base
Cache API Entries	NumberOfItems32	Total number of entries within the cache added by the user.
Cache API Turnover Rate	RateOfCountsPerSecond32	Number of additions and removals to the API cache per second.
Cache API Hits	NumberOfItems32	Number of cache hits from user code.
Cache API Misses	NumberOfItems32	Number of cache misses called from user code.
Cache API Hit Ratio	RawFraction	Ratio of hits called from user code.
Cache API Hit Ratio Base	RawBase	Cache API Hit Ratio Base
Output Cache Entries	NumberOfItems32	Current number of entries in the output cache.
Output Cache Turnover Rate	RateOfCountsPerSecond32	Number of additions and removals to the output cache per second.
Output Cache Hits	NumberOfItems32	Total number of output cacheable requests served from the output cache.
Output Cache Misses	NumberOfItems32	Total number of output cacheable requests not served from the output cache.
Output Cache Hit Ratio	RawFraction	Ratio of hits to requests for output cacheable requests.
Output Cache Hit Ratio Base	RawBase	Output Cache Hit Ratio Base
Compilations Total	NumberOfItems32	Number of .aspx, .ascx, .ashx, .asmx, or .aspx source files dynamically compiled.
Debugging Requests	NumberOfItems32	Number of debugging requests processed.
Errors During Preprocessing	NumberOfItems32	Number of errors that have occurred during parsing and

Counter Name	Counter Type	Counter Description
		configuration.
Errors During Compilation	NumberOfItems32	Number of errors that have occurred during compilation.
Errors During Execution	NumberOfItems32	Number of errors that have occurred during the processing of a request.
Errors Unhandled During Execution	NumberOfItems32	Number of errors not handled by user code, but by the default error handler.
Errors Unhandled During Execution/Sec	RateOfCountsPerSecond32	Rate of unhandled errors.
Errors Total	NumberOfItems32	Total number of errors occurred.
Errors Total/Sec	RateOfCountsPerSecond32	Rate of errors occurred.
Pipeline Instance Count	NumberOfItems32	Number of active pipeline instances.
Request Bytes In Total	NumberOfItems32	The total size, in bytes, of all requests.
Request Bytes Out Total	NumberOfItems32	The total size, in bytes, of responses sent to a client. This does not include standard HTTP response headers.
Requests Executing	NumberOfItems32	The number of requests currently executing.
Requests Failed	NumberOfItems32	Total number of failed requests.
Requests Not Found	NumberOfItems32	The number of requests for resources that were not found.
Requests Not Authorized	NumberOfItems32	Number of requests failed due to unauthorized access.
Requests In Application Queue	NumberOfItems32	The number of requests in the application request queue.

Counter Name	Counter Type	Counter Description
Requests Timed Out	NumberOfItems32	The number of requests that timed out.
Requests Succeeded	NumberOfItems32	The number of requests that executed successfully.
Requests Total	NumberOfItems32	The total number of requests since the application was started.
Requests/Sec	RateOfCountsPerSecond32	The number of requests executed per second.
Sessions Active	NumberOfItems32	The current number of sessions currently active.
Sessions Abandoned	NumberOfItems32	The number of sessions that have been explicitly abandoned.
Sessions Timed Out	NumberOfItems32	The number of sessions timed out.
Sessions Total	NumberOfItems32	Total number of sessions since the application was started.
Transactions Aborted	NumberOfItems32	The number of transactions aborted.
Transactions Committed	NumberOfItems32	The number of transactions committed.
Transactions Pending	NumberOfItems32	Number of transactions in progress.
Transactions Total	NumberOfItems32	The total number of transactions since the application was started.
Transactions/Sec	RateOfCountsPerSecond32	Transactions started per second.
Session State Server connections total	NumberOfItems32	The total number of connections to the State Server used by session state.
Session SQL Server connections total	NumberOfItems32	The total number of connections to the SQL Server used by session state.

Counter Name	Counter Type	Counter Description
Events Raised	NumberOfItems32	Total number of instrumentation events raised since the application was started.
Events Raised/Sec	RateOfCountsPerSecond32	Total number of instrumentation events per second.
Application Lifetime Events	NumberOfItems32	Number of application events raised since the application was started.
Application Lifetime Events/Sec	RateOfCountsPerSecond32	Number of application events raised per second.
Error Events Raised	NumberOfItems32	Number of error events raised since the application was started.
Error Events Raised/Sec	RateOfCountsPerSecond32	Number of error events per second.
Request Error Events Raised	NumberOfItems32	Number of runtime error events raised since the application was started.
Request Error Events Raised/Sec	RateOfCountsPerSecond32	Number of runtime error events per second.
Infrastructure Error Events Raised	NumberOfItems32	Number of HTTP error events raised since the application was started.
Infrastructure Error Events Raised/Sec	RateOfCountsPerSecond32	Number of HTTP error events raised per second.
Request Events Raised	NumberOfItems32	Number of request events raised since the application was started
Request Events Raised/Sec	RateOfCountsPerSecond32	Number of request events raised per second.
Audit Success Events Raised	NumberOfItems32	Number of audit successes in the application since it was started.
Audit Failure Events Raised	NumberOfItems32	Number of audit failures in the application since it was started.
Membership	NumberOfItems32	Number of successful membership credential validations

Counter Name	Counter Type	Counter Description
Authentication Success		since the application was started.
Membership Authentication Failure	NumberOfItems32	Number of failed membership credential validations since the application was started.
Forms Authentication Success	NumberOfItems32	Number of successful forms authentication ticket validations since the application was started.
Forms Authentication Failure	NumberOfItems32	Number of failed forms authentication ticket validations since the application was started.
Viewstate MAC Validation Failure	NumberOfItems32	Number of viewstate MAC validations that failed since the application was started.
Request Execution Time	NumberOfItems32	The number of milliseconds that it took to execute the most recent request.
Requests Disconnected	NumberOfItems32	The number of requests disconnected due to communication errors or user terminated.
Requests Rejected	NumberOfItems32	The number of requests rejected because the application request queue was full.
Request Wait Time	NumberOfItems32	The number of milliseconds the most recent request was waiting in the queue.
Cache % Machine Memory Limit Used	RawFraction	The amount of physical memory used by the machine divided by the physical memory limit for the cache, as a percentage. When this reaches 100%, half of the cache entries will be forcibly removed. The <code>__Total__</code> instance is the average of all instances, and therefore cannot be used to determine when cache entries will be forcibly removed.
Cache % Machine Memory Limit Used Base	RawBase	Cache % Machine Memory Limit Used Base
Cache % Process Memory Limit Used	RawFraction	The value of private bytes for the worker process divided by the private bytes memory limit for the cache, as a percentage. When this reaches 100%, half of the cache entries will be forcibly removed. The <code>__Total__</code> instance

Counter Name	Counter Type	Counter Description
		is the average of all instances, and therefore cannot be used to determine when cache entries will be forcibly removed.
Cache % Process Memory Limit Used Base	RawBase	Cache % Process Memory Limit Used Base
Cache Total Trims	NumberOfItems32	Total number of entries forcibly removed from the cache due to memory pressure.
Cache API Trims	NumberOfItems32	Total number of entries forcibly removed from the cache due to memory pressure that were originally inserted into the cache using one of the public cache APIs.
Output Cache Trims	NumberOfItems32	Total number of entries forcibly removed from the cache due to memory pressure that were originally inserted into the cache by the output cache feature.

Category: ASP.NET Apps v4.0.30319

CategoryType: MultiInstance.

ASP.NET application performance counters

Instance: Total

Counter Name	Counter Type	Counter Description
Anonymous Requests	NumberOfItems32	Number of requests utilizing anonymous authentication.
Anonymous Requests/Sec	RateOfCountsPerSecond32	Number of Authentication Anonymous Requests/Sec
Cache Total Entries	NumberOfItems32	Total number of entries within the cache (both internal and user added)
Cache Total Turnover Rate	RateOfCountsPerSecond32	Number of additions and removals to the total cache per second.
Cache Total Hits	NumberOfItems32	Total number of hits from the cache.

Counter Name	Counter Type	Counter Description
Cache Total Misses	NumberOfItems32	Total number of cache misses.
Cache Total Hit Ratio	RawFraction	Ratio of hits from all cache calls.
Cache Total Hit Ratio Base	RawBase	Cache Total Hit Ratio Base
Cache API Entries	NumberOfItems32	Total number of entries within the cache added by the user.
Cache API Turnover Rate	RateOfCountsPerSecond32	Number of additions and removals to the API cache per second.
Cache API Hits	NumberOfItems32	Number of cache hits from user code.
Cache API Misses	NumberOfItems32	Number of cache misses called from user code.
Cache API Hit Ratio	RawFraction	Ratio of hits called from user code.
Cache API Hit Ratio Base	RawBase	Cache API Hit Ratio Base
Output Cache Entries	NumberOfItems32	Current number of entries in the output cache.
Output Cache Turnover Rate	RateOfCountsPerSecond32	Number of additions and removals to the output cache per second.
Output Cache Hits	NumberOfItems32	Total number of output cacheable requests served from the output cache.
Output Cache Misses	NumberOfItems32	Total number of output cacheable requests not served from the output cache.
Output Cache Hit Ratio	RawFraction	Ratio of hits to requests for output cacheable requests.
Output Cache Hit Ratio Base	RawBase	Output Cache Hit Ratio Base
Compilations Total	NumberOfItems32	Number of .asax, .ascx, .ashx, .asmx, or .aspx source

Counter Name	Counter Type	Counter Description
		files dynamically compiled.
Debugging Requests	NumberOfItems32	Number of debugging requests processed.
Errors During Preprocessing	NumberOfItems32	Number of errors that have occurred during parsing and configuration.
Errors During Compilation	NumberOfItems32	Number of errors that have occurred during compilation.
Errors During Execution	NumberOfItems32	Number of errors that have occurred during the processing of a request.
Errors Unhandled During Execution	NumberOfItems32	Number of errors not handled by user code, but by the default error handler.
Errors Unhandled During Execution/Sec	RateOfCountsPerSecond32	Rate of unhandled errors.
Errors Total	NumberOfItems32	Total number of errors occurred.
Errors Total/Sec	RateOfCountsPerSecond32	Rate of errors occurred.
Pipeline Instance Count	NumberOfItems32	Number of active pipeline instances.
Request Bytes In Total	NumberOfItems32	The total size, in bytes, of all requests.
Request Bytes Out Total	NumberOfItems32	The total size, in bytes, of responses sent to a client. This does not include standard HTTP response headers.
Requests Executing	NumberOfItems32	The number of requests currently executing.
Requests Failed	NumberOfItems32	Total number of failed requests.
Requests Not Found	NumberOfItems32	The number of requests for resources that were not found.
Requests Not Authorized	NumberOfItems32	Number of requests failed due to unauthorized access.

Counter Name	Counter Type	Counter Description
Requests In Application Queue	NumberOfItems32	The number of requests in the application request queue.
Requests Timed Out	NumberOfItems32	The number of requests that timed out.
Requests Succeeded	NumberOfItems32	The number of requests that executed successfully.
Requests Total	NumberOfItems32	The total number of requests since the application was started.
Requests/Sec	RateOfCountsPerSecond32	The number of requests executed per second.
Sessions Active	NumberOfItems32	The current number of sessions currently active.
Sessions Abandoned	NumberOfItems32	The number of sessions that have been explicitly abandoned.
Sessions Timed Out	NumberOfItems32	The number of sessions timed out.
Sessions Total	NumberOfItems32	Total number of sessions since the application was started.
Transactions Aborted	NumberOfItems32	The number of transactions aborted.
Transactions Committed	NumberOfItems32	The number of transactions committed.
Transactions Pending	NumberOfItems32	Number of transactions in progress.
Transactions Total	NumberOfItems32	The total number of transactions since the application was started.
Transactions/Sec	RateOfCountsPerSecond32	Transactions started per second.
Session State Server connections total	NumberOfItems32	The total number of connections to the State Server used by session state.

Counter Name	Counter Type	Counter Description
Session SQL Server connections total	NumberOfItems32	The total number of connections to the SQL Server used by session state.
Events Raised	NumberOfItems32	Total number of instrumentation events raised since the application was started.
Events Raised/Sec	RateOfCountsPerSecond32	Total number of instrumentation events per second.
Application Lifetime Events	NumberOfItems32	Number of application events raised since the application was started.
Application Lifetime Events/Sec	RateOfCountsPerSecond32	Number of application events raised per second.
Error Events Raised	NumberOfItems32	Number of error events raised since the application was started.
Error Events Raised/Sec	RateOfCountsPerSecond32	Number of error events per second.
Request Error Events Raised	NumberOfItems32	Number of runtime error events raised since the application was started.
Request Error Events Raised/Sec	RateOfCountsPerSecond32	Number of runtime error events per second.
Infrastructure Error Events Raised	NumberOfItems32	Number of HTTP error events raised since the application was started.
Infrastructure Error Events Raised/Sec	RateOfCountsPerSecond32	Number of HTTP error events raised per second.
Request Events Raised	NumberOfItems32	Number of request events raised since the application was started
Request Events Raised/Sec	RateOfCountsPerSecond32	Number of request events raised per second.
Audit Success Events Raised	NumberOfItems32	Number of audit successes in the application since it was started.

Counter Name	Counter Type	Counter Description
Audit Failure Events Raised	NumberOfItems32	Number of audit failures in the application since it was started.
Membership Authentication Success	NumberOfItems32	Number of successful membership credential validations since the application was started.
Membership Authentication Failure	NumberOfItems32	Number of failed membership credential validations since the application was started.
Forms Authentication Success	NumberOfItems32	Number of successful forms authentication ticket validations since the application was started.
Forms Authentication Failure	NumberOfItems32	Number of failed forms authentication ticket validations since the application was started.
Viewstate MAC Validation Failure	NumberOfItems32	Number of viewstate MAC validations that failed since the application was started.
Request Execution Time	NumberOfItems32	The number of milliseconds that it took to execute the most recent request.
Requests Disconnected	NumberOfItems32	The number of requests disconnected due to communication errors or user terminated.
Requests Rejected	NumberOfItems32	The number of requests rejected because the application request queue was full.
Request Wait Time	NumberOfItems32	The number of milliseconds the most recent request was waiting in the queue.
Cache % Machine Memory Limit Used	RawFraction	The amount of physical memory used by the machine divided by the physical memory limit for the cache, as a percentage. When this reaches 100%, half of the cache entries will be forcibly removed. The <code>__Total__</code> instance is the average of all instances, and therefore cannot be used to determine when cache entries will be forcibly removed.
Cache % Machine Memory Limit Used Base	RawBase	Cache % Machine Memory Limit Used Base
Cache % Process	RawFraction	The value of private bytes for the worker process divided

Counter Name	Counter Type	Counter Description
Memory Limit Used		by the private bytes memory limit for the cache, as a percentage. When this reaches 100%, half of the cache entries will be forcibly removed. The <code>__Total__</code> instance is the average of all instances, and therefore cannot be used to determine when cache entries will be forcibly removed.
Cache % Process Memory Limit Used Base	RawBase	Cache % Process Memory Limit Used Base
Cache Total Trims	NumberOfItems32	Total number of entries forcibly removed from the cache due to memory pressure.
Cache API Trims	NumberOfItems32	Total number of entries forcibly removed from the cache due to memory pressure that were originally inserted into the cache using one of the public cache APIs.
Output Cache Trims	NumberOfItems32	Total number of entries forcibly removed from the cache due to memory pressure that were originally inserted into the cache by the output cache feature.
% Managed Processor Time (estimated)	RawFraction	Estimated percentage of elapsed time that the processor spends executing managed application code. This counter only tracks processor time of managed threads in the application. It does not include additional processor time spent executing on non-managed threads. Note that this counter is only updated with new data every five seconds.
% Managed Processor Time Base (estimated)	RawBase	% Managed Processor Time Base (estimated)
Managed Memory Used (estimated)	NumberOfItems32	Estimated managed heap memory consumption (in KB) by the application. The accuracy of this counter varies depending on the duration of elapsed time since the last full managed memory heap collection. Note that this counter is only updated with new data every five seconds.

Category: ASP.NET State Service

CategoryType: SingleInstance.

ASP.NET State Service

Counter Name	Counter Type	Counter Description
State Server Sessions Timed Out	NumberOfItems3 2	The number of sessions timed out.
State Server Sessions Total	NumberOfItems3 2	The number of sessions total.
State Server Sessions Active	NumberOfItems3 2	The current number of sessions currently active.
State Server Sessions Abandoned	NumberOfItems3 2	The number of sessions that have been explicitly abandoned.

Category: ASP.NET v2.0.50727

CategoryType: SingleInstance.

ASP.NET global performance counters

Counter Name	Counter Type	Counter Description
State Server Sessions Total	NumberOfItems3 2	The number of sessions total.
Requests Current	NumberOfItems3 2	The current number of requests, including those that are queued, currently executing, or waiting to be written to the client. Under the ASP.NET process model, when this counter exceeds the requestQueueLimit defined in the processModel configuration section, ASP.NET will begin rejecting requests.
State Server Sessions Abandoned	NumberOfItems3 2	The number of sessions that have been explicitly abandoned.
State Server Sessions Timed Out	NumberOfItems3 2	The number of sessions timed out.
Audit Success Events Raised	NumberOfItems3 2	Number of audit successes in the application since it was started.
Request Error Events Raised	NumberOfItems3 2	Number of runtime error events raised since the application was started.

Counter Name	Counter Type	Counter Description
Infrastructure Error Events Raised	NumberOfItems32	Number of HTTP error events raised since the application was started.
Audit Failure Events Raised	NumberOfItems32	Number of audit failures in the application since it was started.
Error Events Raised	NumberOfItems32	Number of error events raised since the application was started.
State Server Sessions Active	NumberOfItems32	The current number of sessions currently active.
Requests Disconnected	NumberOfItems32	The number of requests disconnected due to communication errors or user terminated.
Request Execution Time	NumberOfItems32	The number of milliseconds that it took to execute the most recent request.
Application Restarts	NumberOfItems32	Number of times the application has been restarted during the web server's lifetime.
Applications Running	NumberOfItems32	Number of currently running web applications.
Requests Rejected	NumberOfItems32	The number of requests rejected because the request queue was full.
Worker Process Restarts	NumberOfItems32	Number of times a worker process has restarted on the machine.
Request Wait Time	NumberOfItems32	The number of milliseconds the most recent request was waiting in the queue.
Requests Queued	NumberOfItems32	The number of requests waiting to be processed.
Worker Processes Running	NumberOfItems32	Number of worker processes running on the machine.

Category: ASP.NET v4.0.30319

CategoryType: SingleInstance.

ASP.NET global performance counters

Counter Name	Counter Type	Counter Description
State Server Sessions Total	NumberOfItems32	The number of sessions total.
Requests Current	NumberOfItems32	The current number of requests, including those that are queued, currently executing, or waiting to be written to the client. Under the ASP.NET process model, when this counter exceeds the requestQueueLimit defined in the processModel configuration section, ASP.NET will begin rejecting requests.
State Server Sessions Abandoned	NumberOfItems32	The number of sessions that have been explicitly abandoned.
State Server Sessions Timed Out	NumberOfItems32	The number of sessions timed out.
Audit Success Events Raised	NumberOfItems32	Number of audit successes in the application since it was started.
Request Error Events Raised	NumberOfItems32	Number of runtime error events raised since the application was started.
Infrastructure Error Events Raised	NumberOfItems32	Number of HTTP error events raised since the application was started.
Audit Failure Events Raised	NumberOfItems32	Number of audit failures in the application since it was started.
Error Events Raised	NumberOfItems32	Number of error events raised since the application was started.
State Server Sessions Active	NumberOfItems32	The current number of sessions currently active.
Requests Disconnected	NumberOfItems32	The number of requests disconnected due to communication errors or user terminated.
Request Execution Time	NumberOfItems32	The number of milliseconds that it took to execute the most recent request.
Application Restarts	NumberOfItems32	Number of times the application has been restarted during the web server's lifetime.

Counter Name	Counter Type	Counter Description
Applications Running	NumberOfItems32	Number of currently running web applications.
Requests Rejected	NumberOfItems32	The number of requests rejected because the request queue was full.
Worker Process Restarts	NumberOfItems32	Number of times a worker process has restarted on the machine.
Request Wait Time	NumberOfItems32	The number of milliseconds the most recent request was waiting in the queue.
Requests Queued	NumberOfItems32	The number of requests waiting to be processed.
Worker Processes Running	NumberOfItems32	Number of worker processes running on the machine.

Category: Browser

CategoryType: SingleInstance.

The Browser performance object consists of counters that measure the rates of announcements, enumerations, and other Browser transmissions.

Counter Name	Counter Type	Counter Description
Server Announce Allocations Failed/sec	RateOfCountsPerSecond32	Server Announce Allocations Failed/sec is the rate at which server (or domain) announcements have failed due to lack of memory.
Mailslot Allocations Failed	NumberOfItems32	Mailslot Allocations Failed is the number of times the datagram receiver has failed to allocate a buffer to hold a user mailslot write.
Missed Server List Requests	NumberOfItems32	Missed Server List Requests is the number of requests to retrieve a list of browser servers that were received by this workstation, but could not be processed.
Missed Server Announcements	NumberOfItems32	Missed Server Announcements is the number of server announcements that have been missed due to configuration or allocation limits.
Missed Mailslot Datagrams	NumberOfItems32	Missed Mailslot Datagrams is the number of Mailslot Datagrams that have been discarded due to

Counter Name	Counter Type	Counter Description
		configuration or allocation limits.
Duplicate Master Announcements	NumberOfItems32	Duplicate Master Announcements indicates the number of times that the master browser has detected another master browser on the same domain.
Illegal Datagrams/sec	RateOfCountsPerSecond64	Illegal Datagrams/sec is the rate at which incorrectly formatted datagrams have been received by the workstation.
Mailslot Opens Failed/sec	RateOfCountsPerSecond32	Mailslot Opens Failed/sec indicates the rate at which mailslot messages to be delivered to mailslots that are not present are received by this workstation.
Mailslot Receives Failed	NumberOfItems32	Mailslot Receives Failed indicates the number of mailslot messages that could not be received due to transport failures.
Mailslot Writes Failed	NumberOfItems32	Mailslot Writes Failed is the total number of mailslot messages that have been successfully received, but that could not be written to the mailslot.
Election Packets/sec	RateOfCountsPerSecond32	Election Packets/sec is the rate at which browser election packets have been received by this workstation.
Mailslot Writes/sec	RateOfCountsPerSecond32	Mailslot Writes/sec is the rate at which mailslot messages have been successfully received.
Announcements Total/sec	RateOfCountsPerSecond64	Announcements Total/sec is the sum of Announcements Server/sec and Announcements Domain/sec.
Announcements Server/sec	RateOfCountsPerSecond64	Announcements Server/sec is the rate at which the servers in this domain have announced themselves to this server.
Announcements Domain/sec	RateOfCountsPerSecond64	Announcements Domain/sec is the rate at which a domain has announced itself to the network.
Enumerations Other/sec	RateOfCountsPerSecond32	Enumerations Other/sec is the rate at which browse requests processed by this workstation are not domain or server browse requests.
Enumerations Total/sec	RateOfCountsPerSecond32	Enumerations Total/sec is the rate at which browse requests have been processed by this workstation. This is the sum of Enumerations Server/sec, Enumerations

Counter Name	Counter Type	Counter Description
		Domain/sec, and Enumerations Other/sec.
Enumerations Domain/sec	RateOfCountsPerSecond3 2	Enumerations Domain/sec is the rate at which domain browse requests have been processed by this workstation.
Server List Requests/sec	RateOfCountsPerSecond3 2	Server List Requests/sec is the rate at which requests to retrieve a list of browser servers have been processed by this workstation.
Enumerations Server/sec	RateOfCountsPerSecond3 2	Enumerations Server/sec is the rate at which server browse requests have been processed by this workstation.

Category: Cache

CategoryType: SingleInstance.

The Cache performance object consists of counters that monitor the file system cache, an area of physical memory that stores recently used data as long as possible to permit access to the data without having to read from the disk. Because applications typically use the cache, the cache is monitored as an indicator of application I/O operations. When memory is plentiful, the cache can grow, but when memory is scarce, the cache can become too small to be effective.

Counter Name	Counter Type	Counter Description
Read Aheads/sec	RateOfCountsPerSecond3 2	Read Aheads/sec is the frequency of reads from the file system cache in which the Cache detects sequential access to a file. The read aheads permit the data to be transferred in larger blocks than those being requested by the application, reducing the overhead per access.
Fast Reads/sec	RateOfCountsPerSecond3 2	Fast Reads/sec is the frequency of reads from the file system cache that bypass the installed file system and retrieve the data directly from the cache. Normally, file I/O requests invoke the appropriate file system to retrieve data from a file, but this path permits direct retrieval of data from the cache without file system involvement if the data is in the cache. Even if the data is not in the cache, one invocation of the file system is avoided.
Sync Fast Reads/sec	RateOfCountsPerSecond3 2	Sync Fast Reads/sec is the frequency of reads from the file system cache that bypass the installed file system and retrieve the data directly from the cache. Normally, file I/O requests invoke the appropriate file system to retrieve data from a file, but this path permits direct retrieval of data from the cache without file system involvement if the data is in the cache. Even if the data is not in the cache, one invocation of the file

Counter Name	Counter Type	Counter Description
		system is avoided. If the data is not in the cache, the request (application program call) will wait until the data has been retrieved from disk.
Sync MDL Reads/sec	RateOfCountsPerSecond32	Sync MDL Reads/sec is the frequency of reads from the file system cache that use a Memory Descriptor List (MDL) to access the pages. The MDL contains the physical address of each page in the transfer, thus permitting Direct Memory Access (DMA) of the pages. If the accessed page(s) are not in main memory, the caller will wait for the pages to fault in from the disk.
Async MDL Reads/sec	RateOfCountsPerSecond32	Async MDL Reads/sec is the frequency of reads from the file system cache that use a Memory Descriptor List (MDL) to access the pages. The MDL contains the physical address of each page in the transfer, thus permitting Direct Memory Access (DMA) of the pages. If the accessed page(s) are not in main memory, the calling application program will not wait for the pages to fault in from disk.
MDL Read Hits %	SampleFraction	MDL Read Hits is the percentage of Memory Descriptor List (MDL) Read requests to the file system cache that hit the cache, i.e., did not require disk accesses in order to provide memory access to the page(s) in the cache.
Async Fast Reads/sec	RateOfCountsPerSecond32	Async Fast Reads/sec is the frequency of reads from the file system cache that bypass the installed file system and retrieve the data directly from the cache. Normally, file I/O requests will invoke the appropriate file system to retrieve data from a file, but this path permits data to be retrieved from the cache directly (without file system involvement) if the data is in the cache. Even if the data is not in the cache, one invocation of the file system is avoided. If the data is not in the cache, the request (application program call) will not wait until the data has been retrieved from disk, but will get control immediately.
Lazy Write Pages/sec	RateOfCountsPerSecond32	Lazy Write Pages/sec is the rate at which the Lazy Writer thread has written to disk. Lazy Writing is the process of updating the disk after the page has been changed in memory, so that the application that changed the file does not have to wait for the disk write to be complete before proceeding. More than one page can be transferred on a single disk write operation.
Data Flushes/sec	RateOfCountsPerSecond32	Data Flushes/sec is the rate at which the file system cache has flushed its contents to disk as the result of a request to flush or to satisfy a write-through file write request. More than one page can be transferred on each flush operation.

Counter Name	Counter Type	Counter Description
Data Flush Pages/sec	RateOfCountsPerSecond32	Data Flush Pages/sec is the number of pages the file system cache has flushed to disk as a result of a request to flush or to satisfy a write-through file write request. More than one page can be transferred on each flush operation.
Fast Read Resource Misses/sec	RateOfCountsPerSecond32	Fast Read Resource Misses/sec is the frequency of cache misses necessitated by the lack of available resources to satisfy the request.
Fast Read Not Possibles/sec	RateOfCountsPerSecond32	Fast Read Not Possibles/sec is the frequency of attempts by an Application Program Interface (API) function call to bypass the file system to get to data in the file system cache that could not be honored without invoking the file system.
Lazy Write Flushes/sec	RateOfCountsPerSecond32	Lazy Write Flushes/sec is the rate at which the Lazy Writer thread has written to disk. Lazy Writing is the process of updating the disk after the page has been changed in memory, so that the application that changed the file does not have to wait for the disk write to be complete before proceeding. More than one page can be transferred by each write operation.
MDL Reads/sec	RateOfCountsPerSecond32	MDL Reads/sec is the frequency of reads from the file system cache that use a Memory Descriptor List (MDL) to access the data. The MDL contains the physical address of each page involved in the transfer, and thus can employ a hardware Direct Memory Access (DMA) device to effect the copy. The LAN Server uses this method for large transfers out of the server.
Data Map Hits %	SampleFraction	Data Map Hits is the percentage of data maps in the file system cache that could be resolved without having to retrieve a page from the disk, because the page was already in physical memory.
Data Map Pins/sec	SampleFraction	Data Map Pins/sec is the frequency of data maps in the file system cache that resulted in pinning a page in main memory, an action usually preparatory to writing to the file on disk. While pinned, a page's physical address in main memory and virtual address in the file system cache will not be altered.
Pin Reads/sec	RateOfCountsPerSecond32	Pin Reads/sec is the frequency of reading data into the file system cache preparatory to writing the data back to disk. Pages read in this fashion are pinned in memory at the completion of the read. While pinned, a page's physical address in the file system cache will not be altered.
Data Maps/sec	RateOfCountsPerSecond32	Data Maps/sec is the frequency that a file system such as NTFS, maps a page of a file into the file system cache to read the page.

Counter Name	Counter Type	Counter Description
Sync Data Maps/sec	RateOfCountsPerSecond32	Sync Data Maps/sec counts the frequency that a file system, such as NTFS, maps a page of a file into the file system cache to read the page, and wishes to wait for the page to be retrieved if it is not in main memory.
Async Data Maps/sec	RateOfCountsPerSecond32	Async Data Maps/sec is the frequency that an application using a file system, such as NTFS, to map a page of a file into the file system cache to read the page, and does not wait for the page to be retrieved if it is not in main memory.
Sync Pin Reads/sec	RateOfCountsPerSecond32	Sync Pin Reads/sec is the frequency of reading data into the file system cache preparatory to writing the data back to disk. Pages read in this fashion are pinned in memory at the completion of the read. The file system will not regain control until the page is pinned in the file system cache, in particular if the disk must be accessed to retrieve the page. While pinned, a page's physical address in the file system cache will not be altered.
Sync Copy Reads/sec	RateOfCountsPerSecond32	Sync Copy Reads/sec is the frequency of reads from pages of the file system cache that involve a memory copy of the data from the cache to the application's buffer. The file system will not regain control until the copy operation is complete, even if the disk must be accessed to retrieve the page.
Async Copy Reads/sec	RateOfCountsPerSecond32	Async Copy Reads/sec is the frequency of reads from pages of the file system cache that involve a memory copy of the data from the cache to the application's buffer. The application will regain control immediately even if the disk must be accessed to retrieve the page.
Copy Read Hits %	SampleFraction	Copy Read Hits is the percentage of cache copy read requests that hit the cache, that is, they did not require a disk read in order to provide access to the page in the cache. A copy read is a file read operation that is satisfied by a memory copy from a page in the cache to the application's buffer. The LAN Redirector uses this method for retrieving information from the cache, as does the LAN Server for small transfers. This is a method used by the disk file systems as well.
Async Pin Reads/sec	RateOfCountsPerSecond32	Async Pin Reads/sec is the frequency of reading data into the file system cache preparatory to writing the data back to disk. Pages read in this fashion are pinned in memory at the completion of the read. The file system will regain control immediately even if the disk must be accessed to retrieve the page. While pinned, a page's physical address will not be altered.
Pin Read Hits %	SampleFraction	Pin Read Hits is the percentage of pin read requests that hit the file system cache, i.e., did not require a disk read in order to

Counter Name	Counter Type	Counter Description
		provide access to the page in the file system cache. While pinned, a page's physical address in the file system cache will not be altered. The LAN Redirector uses this method for retrieving data from the cache, as does the LAN Server for small transfers. This is usually the method used by the disk file systems as well.
Copy Reads/sec	RateOfCountsPerSecond3 2	Copy Reads/sec is the frequency of reads from pages of the file system cache that involve a memory copy of the data from the cache to the application's buffer. The LAN Redirector uses this method for retrieving information from the file system cache, as does the LAN Server for small transfers. This is a method used by the disk file systems as well.

Category: Distributed Transaction Coordinator

CategoryType: SingleInstance.

Microsoft Distributed Transaction Coordinator performance counters

Counter Name	Counter Type	Counter Description
Response Time -- Maximum	NumberOfItems3 2	Maximum time delta between transaction begin and commit
Response Time -- Average	NumberOfItems3 2	Average time delta between transaction begin and commit
Response Time -- Minimum	NumberOfItems3 2	Minimum time delta between transaction begin and commit
Aborted Transactions/sec	NumberOfItems3 2	Transactions aborted per second
Committed Transactions/sec	NumberOfItems3 2	Transactions committed per second
Transactions/sec	NumberOfItems3 2	Transactions performed per second
Force Aborted Transactions	NumberOfItems3 2	Number of transactions aborted by the system administrator
Aborted Transactions	NumberOfItems3 2	Number of aborted transactions

Counter Name	Counter Type	Counter Description
Committed Transactions	NumberOfItems32	Number of committed transactions
Active Transactions	NumberOfItems32	Number of currently active transactions
Force Committed Transactions	NumberOfItems32	Number of transactions committed by the system administrator
Active Transactions Maximum	NumberOfItems32	Maximum number of transactions ever concurrently active
In Doubt Transactions	NumberOfItems32	Number of in doubt transactions

Category: ICMP

CategoryType: SingleInstance.

The ICMP performance object consists of counters that measure the rates at which messages are sent and received by using ICMP protocols. It also includes counters that monitor ICMP protocol errors.

Counter Name	Counter Type	Counter Description
Sent Time Exceeded	NumberOfItems32	Sent Time Exceeded is the number of ICMP Time Exceeded messages sent.
Sent Parameter Problem	NumberOfItems32	Sent Parameter Problem is the number of ICMP Parameter Problem messages sent.
Sent Source Quench	NumberOfItems32	Sent Source Quench is the number of ICMP Source Quench messages sent.
Messages Sent/sec	RateOfCountsPerSecond32	Messages Sent/sec is the rate, in incidents per second, at which the server attempted to send. The rate includes those messages sent in error.
Messages Outbound Errors	NumberOfItems32	Messages Outbound Errors is the number of ICMP messages that were not send due to problems within ICMP, such as lack of buffers. This value does not include errors discovered outside the ICMP layer, such as those recording the failure of IP to route the resultant datagram. In some implementations, none of the error types are included in the value of this counter.

Counter Name	Counter Type	Counter Description
Sent Destination Unreachable	NumberOfItems32	Sent Destination Unreachable is the number of ICMP Destination Unreachable messages sent.
Sent Redirect/sec	RateOfCountsPerSecond32	Sent Redirect/sec is the rate, in incidents per second, at which ICMP Redirect messages were sent.
Sent Timestamp Reply/sec	RateOfCountsPerSecond32	Sent Timestamp Reply/sec is the rate, in incidents per second, at which ICMP Timestamp Reply messages were sent.
Sent Address Mask	NumberOfItems32	Sent Address Mask is the number of ICMP Address Mask Request messages sent.
Sent Address Mask Reply	NumberOfItems32	Sent Address Mask Reply is the number of ICMP Address Mask Reply messages sent.
Sent Echo/sec	RateOfCountsPerSecond32	Sent Echo/sec is the rate of ICMP Echo messages sent.
Sent Echo Reply/sec	RateOfCountsPerSecond32	Sent Echo Reply/sec is the rate, in incidents per second, at which ICMP Echo Reply messages were sent.
Sent Timestamp/sec	RateOfCountsPerSecond32	Sent Timestamp/sec is the rate, in incidents per second, at which ICMP Timestamp Request messages were sent.
Received Address Mask Reply	NumberOfItems32	Received Address Mask Reply is the number of ICMP Address Mask Reply messages received.
Received Dest. Unreachable	NumberOfItems32	Received Destination Unreachable is the number of ICMP Destination Unreachable messages received.
Received Time Exceeded	NumberOfItems32	Received Time Exceeded is the number of ICMP Time Exceeded messages received.
Received Parameter Problem	NumberOfItems32	Received Parameter Problem is the number of ICMP Parameter Problem messages received.
Messages/sec	RateOfCountsPerSecond32	Messages/sec is the total rate, in incidents per second, at which ICMP messages were sent and received by the entity. The rate includes messages received or sent in error.

Counter Name	Counter Type	Counter Description
Messages Received/sec	RateOfCountsPerSecond32	Messages Received/sec is the rate, in incidents per second at which ICMP messages were received. The rate includes messages received in error.
Messages Received Errors	NumberOfItems32	Messages Received Errors is the number of ICMP messages that the entity received but had errors, such as bad ICMP checksums, bad length, etc.
Received Source Quench	NumberOfItems32	Received Source Quench is the number of ICMP Source Quench messages received.
Received Timestamp/sec	RateOfCountsPerSecond32	Received Timestamp/sec is the rate, in incidents per second at which ICMP Timestamp Request messages were received.
Received Timestamp Reply/sec	RateOfCountsPerSecond32	Received Timestamp Reply/sec is the rate of ICMP Timestamp Reply messages received.
Received Address Mask	NumberOfItems32	Received Address Mask is the number of ICMP Address Mask Request messages received.
Received Redirect/sec	RateOfCountsPerSecond32	Received Redirect/sec is the rate, in incidents per second, at which ICMP Redirect messages were received.
Received Echo/sec	RateOfCountsPerSecond32	Received Echo/sec is the rate, in incidents per second, at which ICMP Echo messages were received.
Received Echo Reply/sec	RateOfCountsPerSecond32	Received Echo Reply/sec is the rate, in incidents per second, at which ICMP Echo Reply messages were received.

Category: IP

CategoryType: SingleInstance.

The IP performance object consists of counters that measure the rates at which IP datagrams are sent and received by using IP protocols. It also includes counters that monitor IP protocol errors.

Counter Name	Counter Type	Counter Description
Fragments Received/sec	RateOfCountsPerSecond32	Fragments Received/sec is the rate, in incidents per second, at which IP fragments that need to be reassembled at this entity are received.

Counter Name	Counter Type	Counter Description
Fragments Re-assembled/sec	RateOfCountsPerSecond32	Fragments Re-assembled/sec is the rate, in incidents per second, at which IP fragments were successfully reassembled.
Datagrams Outbound Discarded	NumberOfItems32	Datagrams Outbound Discarded is the number of output IP datagrams that were discarded even though no problems were encountered to prevent their transmission to their destination (for example, lack of buffer space). This counter includes datagrams counted in Datagrams Forwarded/sec that meet this criterion.
Datagrams Outbound No Route	NumberOfItems32	Datagrams Outbound No Route is the number of IP datagrams that were discarded because no route could be found to transmit them to their destination. This counter includes any packets counted in Datagrams Forwarded/sec that meet this 'no route' criterion.
Fragmentation Failures	NumberOfItems32	Fragmentation Failures is the number of IP datagrams that were discarded because they needed to be fragmented at but could not be (for example, because the 'Don't Fragment' flag was set).
Fragments Created/sec	RateOfCountsPerSecond32	Fragments Created/sec is the rate, in incidents per second, at which IP datagram fragments were generated as a result of fragmentation.
Fragment Re-assembly Failures	NumberOfItems32	Fragment Re-assembly Failures is the number of failures detected by the IP reassembly algorithm, such as time outs, errors, etc. This is not necessarily a count of discarded IP fragments since some algorithms (notably RFC 815) lose track of the number of fragments by combining them as they are received.
Fragmented Datagrams/sec	RateOfCountsPerSecond32	Fragmented Datagrams/sec is the rate, in incidents per second, at which datagrams are successfully fragmented.
Datagrams Sent/sec	RateOfCountsPerSecond32	Datagrams Sent/sec is the rate, in incidents per second, at which IP datagrams were supplied for transmission by local IP user-protocols (including ICMP). This counter does not include any datagrams counted in Datagrams Forwarded/sec. Datagrams Sent/sec is a subset of Datagrams/sec.
Datagrams Received Header Errors	NumberOfItems32	Datagrams Received Header Errors is the number of input datagrams that were discarded due to errors in the IP headers, including bad checksums, version number mismatch, other format errors, time-to-live exceeded, errors discovered in processing their IP options, etc.

Counter Name	Counter Type	Counter Description
Datagrams Received Address Errors	NumberOfItems32	Datagrams Received Address Errors is the number of input datagrams that were discarded because the IP address in their IP header destination field was not valid for the computer. This count includes invalid addresses (for example, 0.0. 0.0) and addresses of unsupported Classes (for example, Class E). For entities that are not IP gateways and do not forward datagrams, this counter includes datagrams that were discarded because the destination address was not a local address.
Datagrams/sec	RateOfCountsPerSecond32	Datagrams/sec is the rate, in incidents per second, at which IP datagrams were received from or sent to the interfaces, including those in error. Forwarded datagrams are not included in this rate.
Datagrams Received/sec	RateOfCountsPerSecond32	Datagrams Received/sec is the rate, in incidents per second, at which IP datagrams are received from the interfaces, including those in error. Datagrams Received/sec is a subset of Datagrams/sec.
Datagrams Received Discarded	NumberOfItems32	Datagrams Received Discarded is the number of input IP datagrams that were discarded even though problems prevented their continued processing (for example, lack of buffer space). This counter does not include any datagrams discarded while awaiting re-assembly.
Datagrams Received Delivered/sec	RateOfCountsPerSecond32	Datagrams Received Delivered/sec is the rate, in incidents per second, at which input datagrams were successfully delivered to IP user-protocols, including Internet Control Message Protocol (ICMP).
Datagrams Forwarded/sec	RateOfCountsPerSecond32	Datagrams Forwarded/sec is the rate, in incidents per second, at which attempts were made to find routes to forward input datagrams their final destination, because the local server was not the final IP destination. In servers that do not act as IP Gateways, this rate includes only packets that were source-routed via this entity, where the source-route option processing was successful.
Datagrams Received Unknown Protocol	NumberOfItems32	Datagrams Received Unknown Protocol is the number of locally-addressed datagrams that were successfully received but were discarded because of an unknown or unsupported protocol.

Category: Job Object

CategoryType: MultiInstance.

Reports the accounting and processor usage data collected by each active named Job object.

Instance: WmiProviderSubSystemHostJob

Counter Name	Counter Type	Counter Description
Current % Processor Time	Timer100Ns	Current % Processor Time shows the percentage of the sample interval that the processes in the Job object spent executing code.
Current % User Mode Time	Timer100Ns	Current % User mode Time shows the percentage of the sample interval that the processes in the Job object spent executing code in user mode.
Current % Kernel Mode Time	Timer100Ns	Current % Kernel mode Time shows the percentage of the sample interval that the processes in the Job object spent executing code in kernel or privileged mode.
This Period mSec - Processor	NumberOfItems64	This Period mSec - Processor shows the time, in milliseconds, of processor time used by all the processes in the Job object, including those that have terminated or that are no longer associated with the Job object, since a time limit on the Job was established.
This Period mSec - User Mode	NumberOfItems64	This Period mSec - User mode shows the time, in milliseconds, of user mode processor time used by all the processes in the Job object, including those that have terminated or that are no longer associated with the Job object, since a time limit on the Job was established.
This Period mSec - Kernel Mode	NumberOfItems64	This Period mSec - Kernel mode shows the time, in milliseconds, of kernel mode processor time used by all the processes in the Job object, including those that have terminated or that are no longer associated with the Job object, since a time limit on the Job was established.
Total mSec - Processor	NumberOfItems64	Total mSec - Processor shows the time, in milliseconds, of processor time used by all the processes in the Job object, including those that have terminated or that are no longer associated with the Job object, since the Job object was created.
Total mSec - User Mode	NumberOfItems64	Total mSec - User mode shows the time, in milliseconds, of user mode processor time used by all the processes in the Job object, including those that have terminated or that are no longer associated with the Job object, since the Job object was created.
Total mSec - Kernel Mode	NumberOfItems64	Total mSec - Kernel mode shows the time, in milliseconds, of kernel mode processor time used by all the processes in the Job object, including those that have terminated or that are no longer associated with the Job object, since the Job

Counter Name	Counter Type	Counter Description
		object was created.
Pages/Sec	RateOfCountsPerSecond32	Pages/Sec shows the page fault rate of all the processes in the Job object.
Process Count - Total	NumberOfItems32	Process Count - Total shows the number of processes, both active and terminated, that are or have been associated with the Job object.
Process Count - Active	NumberOfItems32	Process Count - Active shows the number of processes that are currently associated with the Job object.
Process Count - Terminated	NumberOfItems32	Process Count - Terminated shows the number of processes that have been terminated because of a limit violation.

Instance: _Total

Instance: Winlogon Job 0-5895c

Category: Job Object Details

CategoryType: MultiInstance.

% Job object Details shows detailed performance information about the active processes that make up a Job object.

Instance: Winlogon Job 0-5895c/_Total

Counter Name	Counter Type	Counter Description
% Processor Time	Timer100Ns	% Processor Time is the percentage of elapsed time that all of process threads used the processor to execution instructions. An instruction is the basic unit of execution in a computer, a thread is the object that executes instructions, and a process is the object created when a program is run. Code executed to handle some hardware interrupts and trap conditions are included in this count.
% User Time	Timer100Ns	% User Time is the percentage of elapsed time that the process threads spent executing code in user mode. Applications, environment subsystems, and integral subsystems execute in user mode. Code executing in user mode cannot damage the integrity of the Windows executive, kernel, and device drivers. Unlike some early operating systems, Windows uses process boundaries for subsystem

Counter Name	Counter Type	Counter Description
		protection in addition to the traditional protection of user and privileged modes. Some work done by Windows on behalf of the application might appear in other subsystem processes in addition to the privileged time in the process.
% Privileged Time	Timer100Ns	% Privileged Time is the percentage of elapsed time that the process threads spent executing code in privileged mode. When a Windows system service is called, the service will often run in privileged mode to gain access to system-private data. Such data is protected from access by threads executing in user mode. Calls to the system can be explicit or implicit, such as page faults or interrupts. Unlike some early operating systems, Windows uses process boundaries for subsystem protection in addition to the traditional protection of user and privileged modes. Some work done by Windows on behalf of the application might appear in other subsystem processes in addition to the privileged time in the process.
Virtual Bytes Peak	NumberOfItems64	Virtual Bytes Peak is the maximum size, in bytes, of virtual address space the process has used at any one time. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. However, virtual space is finite, and the process might limit its ability to load libraries.
Virtual Bytes	NumberOfItems64	Virtual Bytes is the current size, in bytes, of the virtual address space the process is using. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. Virtual space is finite, and the process can limit its ability to load libraries.
Page Faults/sec	RateOfCountsPerSecond32	Page Faults/sec is the rate at which page faults by the threads executing in this process are occurring. A page fault occurs when a thread refers to a virtual memory page that is not in its working set in main memory. This may not cause the page to be fetched from disk if it is on the standby list and hence already in main memory, or if it is in use by another process with whom the page is shared.
Working Set Peak	NumberOfItems64	Working Set Peak is the maximum size, in bytes, of the Working Set of this process at any point in time. The Working Set is the set of memory pages touched recently by the threads in the process. If free memory in the computer is above a threshold, pages are left in the Working Set of a process even if they are not in use. When free memory falls below a threshold, pages are trimmed from Working Sets. If they are needed they will then be soft-faulted back into the Working Set before they leave main memory.
Working Set	NumberOfItems64	Working Set is the current size, in bytes, of the Working Set

Counter Name	Counter Type	Counter Description
		of this process. The Working Set is the set of memory pages touched recently by the threads in the process. If free memory in the computer is above a threshold, pages are left in the Working Set of a process even if they are not in use. When free memory falls below a threshold, pages are trimmed from Working Sets. If they are needed they will then be soft-faulted back into the Working Set before leaving main memory.
Page File Bytes Peak	NumberOfItems64	Page File Bytes Peak is the maximum number of bytes this process has used in the paging file(s). Paging files are used to store pages of memory used by the process that are not contained in other files. Paging files are shared by all processes, and the lack of space in paging files can prevent other processes from allocating memory.
Page File Bytes	NumberOfItems64	Page File Bytes is the current number of bytes that this process has used in the paging file(s). Paging files are used to store pages of memory used by the process that are not contained in other files. Paging files are shared by all processes, and the lack of space in paging files can prevent other processes from allocating memory.
Private Bytes	NumberOfItems64	Private Bytes is the current size, in bytes, of memory that this process has allocated that cannot be shared with other processes.
Thread Count	NumberOfItems32	The number of threads currently active in this process. An instruction is the basic unit of execution in a processor, and a thread is the object that executes instructions. Every running process has at least one thread.
Priority Base	NumberOfItems32	The current base priority of this process. Threads within a process can raise and lower their own base priority relative to the process' base priority.
Elapsed Time	ElapsedTime	The total elapsed time, in seconds, that this process has been running.
ID Process	NumberOfItems64	ID Process is the unique identifier of this process. ID Process numbers are reused, so they only identify a process for the lifetime of that process.
Creating Process ID	NumberOfItems64	The Creating Process ID value is the Process ID of the process that created the process. The creating process may have terminated, so this value may no longer identify a running process.

Counter Name	Counter Type	Counter Description
Pool Paged Bytes	NumberOfItems32	Pool Paged Bytes is the size, in bytes, of the paged pool, an area of system memory (physical memory used by the operating system) for objects that can be written to disk when they are not being used. Memory\Pool Paged Bytes is calculated differently than Process\Pool Paged Bytes, so it might not equal Process\Pool Paged Bytes_Total. This counter displays the last observed value only; it is not an average.
Pool Nonpaged Bytes	NumberOfItems32	Pool Nonpaged Bytes is the size, in bytes, of the nonpaged pool, an area of system memory (physical memory used by the operating system) for objects that cannot be written to disk, but must remain in physical memory as long as they are allocated. Memory\Pool Nonpaged Bytes is calculated differently than Process\Pool Nonpaged Bytes, so it might not equal Process\Pool Nonpaged Bytes_Total. This counter displays the last observed value only; it is not an average.
Handle Count	NumberOfItems32	The total number 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.
IO Read Operations/sec	RateOfCountsPerSecond64	The rate at which the process is issuing read I/O operations. This counter counts all I/O activity generated by the process to include file, network and device I/Os.
IO Write Operations/sec	RateOfCountsPerSecond64	The rate at which the process is issuing write I/O operations. This counter counts all I/O activity generated by the process to include file, network and device I/Os.
IO Data Operations/sec	RateOfCountsPerSecond64	The rate at which the process is issuing read and write I/O operations. This counter counts all I/O activity generated by the process to include file, network and device I/Os.
IO Other Operations/sec	RateOfCountsPerSecond64	The rate at which the process is issuing I/O operations that are neither read nor write operations (for example, a control function). This counter counts all I/O activity generated by the process to include file, network and device I/Os.
IO Read Bytes/sec	RateOfCountsPerSecond64	The rate at which the process is reading bytes from I/O operations. This counter counts all I/O activity generated by the process to include file, network and device I/Os.
IO Write Bytes/sec	RateOfCountsPerSecond64	The rate at which the process is writing bytes to I/O operations. This counter counts all I/O activity generated by the process to include file, network and device I/Os.

Counter Name	Counter Type	Counter Description
IO Data Bytes/sec	RateOfCountsPerSecond64	The rate at which the process is reading and writing bytes in I/O operations. This counter counts all I/O activity generated by the process to include file, network and device I/Os.
IO Other Bytes/sec	RateOfCountsPerSecond64	The rate at which the process is issuing bytes to I/O operations that do not involve data such as control operations. This counter counts all I/O activity generated by the process to include file, network and device I/Os.

Instance: `_Total`

Instance: `Winlogon Job 0-5895c/logon.scr`

Instance: `WmiProviderSubSystemHostJob/wmiprvse`

Instance: `WmiProviderSubSystemHostJob/_Total`

Category: LogicalDisk

CategoryType: `MultiInstance`.

The Logical Disk performance object consists of counters that monitor logical partitions of a hard or fixed disk drives. Performance Monitor identifies logical disks by their a drive letter, such as C.

Instance: `_Total`

Counter Name	Counter Type	Counter Description
% Free Space	RawFraction	% Free Space is the percentage of total usable space on the selected logical disk drive that was free.
Free Megabytes	NumberOfItems32	Free Megabytes displays the unallocated space, in megabytes, on the disk drive in megabytes. One megabyte is equal to 1,048,576 bytes.
Current Disk Queue Length	NumberOfItems32	Current Disk Queue Length is the number of requests outstanding on the disk at the time the performance data is collected. It also includes requests in service at the time of the collection. This is a instantaneous snapshot, not an average over the time interval. Multi-spindle disk devices can have multiple requests that are active at one time, but other concurrent requests are awaiting service. This counter might reflect a transitory high or low queue length, but if there is a sustained load on the disk drive, it is likely that this will be consistently high. Requests experience delays proportional to the length of this queue minus the number of spindles on the

Counter Name	Counter Type	Counter Description
		disks. For good performance, this difference should average less than two.
% Disk Time	542573824	% Disk Time is the percentage of elapsed time that the selected disk drive was busy servicing read or write requests.
Avg. Disk Queue Length	5571840	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.
% Disk Read Time	542573824	% Disk Read Time is the percentage of elapsed time that the selected disk drive was busy servicing read requests.
Avg. Disk Read Queue Length	5571840	Avg. Disk Read Queue Length is the average number of read requests that were queued for the selected disk during the sample interval.
% Disk Write Time	542573824	% Disk Write Time is the percentage of elapsed time that the selected disk drive was busy servicing write requests.
Avg. Disk Write Queue Length	5571840	Avg. Disk Write Queue Length is the average number of write requests that were queued for the selected disk during the sample interval.
Avg. Disk sec/Transfer	AverageTimer32	Avg. Disk sec/Transfer is the time, in seconds, of the average disk transfer.
Avg. Disk sec/Read	AverageTimer32	Avg. Disk sec/Read is the average time, in seconds, of a read of data from the disk.
Avg. Disk sec/Write	AverageTimer32	Avg. Disk sec/Write is the average time, in seconds, of a write of data to the disk.
Disk Transfers/sec	RateOfCountsPerSecond32	Disk Transfers/sec is the rate of read and write operations on the disk.
Disk Reads/sec	RateOfCountsPerSecond32	Disk Reads/sec is the rate of read operations on the disk.
Disk Writes/sec	RateOfCountsPerSecond32	Disk Writes/sec is the rate of write operations on the disk.

Counter Name	Counter Type	Counter Description
Disk Bytes/sec	RateOfCountsPerSecond64	Disk Bytes/sec is the rate bytes are transferred to or from the disk during write or read operations.
Disk Read Bytes/sec	RateOfCountsPerSecond64	Disk Read Bytes/sec is the rate at which bytes are transferred from the disk during read operations.
Disk Write Bytes/sec	RateOfCountsPerSecond64	Disk Write Bytes/sec is rate at which bytes are transferred to the disk during write operations.
Avg. Disk Bytes/Transfer	AverageCount64	Avg. Disk Bytes/Transfer is the average number of bytes transferred to or from the disk during write or read operations.
Avg. Disk Bytes/Read	AverageCount64	Avg. Disk Bytes/Read is the average number of bytes transferred from the disk during read operations.
Avg. Disk Bytes/Write	AverageCount64	Avg. Disk Bytes/Write is the average number of bytes transferred to the disk during write operations.
% Idle Time	542573824	% Idle Time reports the percentage of time during the sample interval that the disk was idle.
Split IO/Sec	RateOfCountsPerSecond32	Split IO/Sec reports the rate at which I/Os to the disk were split into multiple I/Os. A split I/O may result from requesting data of a size that is too large to fit into a single I/O or that the disk is fragmented.

Instance: D:

Instance: C:

Category: Memory

CategoryType: SingleInstance.

The Memory performance object consists of counters that describe the behavior of physical and virtual memory on the computer. Physical memory is the amount of random access memory on the computer. Virtual memory consists of the space in physical memory and on disk. Many of the memory counters monitor paging, which is the movement of pages of code and data between disk and physical memory. Excessive paging, a symptom of a memory shortage, can cause delays which interfere with all system processes.

Counter Name	Counter Type	Counter Description
Cache Bytes Peak	NumberOfItems64	Cache Bytes Peak is the maximum number of bytes used by the file system cache since the system was last restarted.

Counter Name	Counter Type	Counter Description
		This might be larger than the current size of the cache. This counter displays the last observed value only; it is not an average.
Pool Paged Resident Bytes	NumberOfItems64	Pool Paged Resident Bytes is the current size, in bytes, of the paged pool. The paged pool is an area of system memory (physical memory used by the operating system) for objects that can be written to disk when they are not being used. Space used by the paged and nonpaged pools are taken from physical memory, so a pool that is too large denies memory space to processes. This counter displays the last observed value only; it is not an average.
System Code Total Bytes	NumberOfItems64	System Code Total Bytes is the size, in bytes, of the pageable operating system code currently in virtual memory. It is a measure of the amount of physical memory being used by the operating system that can be written to disk when not in use. This value is calculated by summing the bytes in Ntoskrnl.exe, Hal.dll, the boot drivers, and file systems loaded by Ntldr/osloader. This counter does not include code that must remain in physical memory and cannot be written to disk. This counter displays the last observed value only; it is not an average.
Cache Bytes	NumberOfItems64	Cache Bytes is the sum of the Memory\System Cache Resident Bytes, Memory\System Driver Resident Bytes, Memory\System Code Resident Bytes, and Memory\Pool Paged Resident Bytes counters. This counter displays the last observed value only; it is not an average.
Pool Paged Allocs	NumberOfItems32	Pool Paged Allocs is the number of calls to allocate space in the paged pool. The paged pool is an area of system memory (physical memory used by the operating system) for objects that can be written to disk when they are not being used. It is measured in numbers of calls to allocate space, regardless of the amount of space allocated in each call. This counter displays the last observed value only; it is not an average.
Pool Nonpaged Allocs	NumberOfItems32	Pool Nonpaged Allocs is the number of calls to allocate space in the nonpaged pool. The nonpaged pool is an area of system memory area for objects that cannot be written to disk, and must remain in physical memory as long as they are allocated. It is measured in numbers of calls to allocate space, regardless of the amount of space allocated in each call. This counter displays the last observed value only; it is not an average.
Free System Page Table Entries	NumberOfItems32	Free System Page Table Entries is the number of page table entries not currently in used by the system. This counter displays the last observed value only; it is not an average.

Counter Name	Counter Type	Counter Description
% Committed Bytes In Use	RawFraction	% Committed Bytes In Use is the ratio of Memory\Committed Bytes to the Memory\Commit Limit. Committed memory is the physical memory in use for which space has been reserved in the paging file should it need to be written to disk. The commit limit is determined by the size of the paging file. If the paging file is enlarged, the commit limit increases, and the ratio is reduced). This counter displays the current percentage value only; it is not an average.
Available KBytes	NumberOfItems64	Available KBytes is the amount of physical memory available to processes running on the computer, in Kilobytes, rather than bytes as reported in Memory\Available Bytes. It is calculated by adding the amount of space on the Zeroed, Free, and Stand by memory lists. Free memory is ready for use; Zeroed memory are pages of memory filled with zeros to prevent later processes from seeing data used by a previous process; Standby memory is memory removed from a process' working set (its physical memory) on route to disk, but is still available to be recalled. This counter displays the last observed value only; it is not an average.
Available MBytes	NumberOfItems64	Available MBytes is the amount of physical memory available to processes running on the computer, in Megabytes, rather than bytes as reported in Memory\Available Bytes. It is calculated by adding the amount of space on the Zeroed, Free, and Stand by memory lists. Free memory is ready for use; Zeroed memory are pages of memory filled with zeros to prevent later processes from seeing data used by a previous process; Standby memory is memory removed from a process' working set (its physical memory) on route to disk, but is still available to be recalled. This counter displays the last observed value only; it is not an average.
System Cache Resident Bytes	NumberOfItems64	System Cache Resident Bytes is the size, in bytes, of the pageable operating system code in the file system cache. This value includes only current physical pages and does not include any virtual memory pages not currently resident. It does equal the System Cache value shown in Task Manager. As a result, this value may be smaller than the actual amount of virtual memory in use by the file system cache. This value is a component of Memory\System Code Resident Bytes which represents all pageable operating system code that is currently in physical memory. This counter displays the last observed value only; it is not an average.
System Code Resident Bytes	NumberOfItems64	System Code Resident Bytes is the size, in bytes of the operating system code currently in physical memory that can be written to disk when not in use. This value is a component of Memory\System Code Total Bytes, which

Counter Name	Counter Type	Counter Description
		also includes operating system code on disk. Memory\\System Code Resident Bytes (and Memory\\System Code Total Bytes) does not include code that must remain in physical memory and cannot be written to disk. This counter displays the last observed value only; it is not an average.
System Driver Total Bytes	NumberOfItems64	System Driver Total Bytes is the size, in bytes, of the pageable virtual memory currently being used by device drivers. Pageable memory can be written to disk when it is not being used. It includes physical memory (Memory\\System Driver Resident Bytes) and code and data paged to disk. It is a component of Memory\\System Code Total Bytes. This counter displays the last observed value only; it is not an average.
System Driver Resident Bytes	NumberOfItems64	System Driver Resident Bytes is the size, in bytes, of the pageable physical memory being used by device drivers. It is the working set (physical memory area) of the drivers. This value is a component of Memory\\System Driver Total Bytes, which also includes driver memory that has been written to disk. Neither Memory\\System Driver Resident Bytes nor Memory\\System Driver Total Bytes includes memory that cannot be written to disk.
Page Writes/sec	RateOfCountsPerSecond3 2	Page Writes/sec is the rate at which pages are written to disk to free up space in physical memory. Pages are written to disk only if they are changed while in physical memory, so they are likely to hold data, not code. This counter shows write operations, without regard to the number of pages written in each operation. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.
Write Copies/sec	RateOfCountsPerSecond3 2	Write Copies/sec is the rate at which page faults are caused by attempts to write that have been satisfied by copying of the page from elsewhere in physical memory. This is an economical way of sharing data since pages are only copied when they are written to; otherwise, the page is shared. This counter shows the number of copies, without regard for the number of pages copied in each operation.
Transition Faults/sec	RateOfCountsPerSecond3 2	Transition Faults/sec is the rate at which page faults are resolved by recovering pages that were being used by another process sharing the page, or were on the modified page list or the standby list, or were being written to disk at the time of the page fault. The pages were recovered without additional disk activity. Transition faults are counted in numbers of faults; because only one page is faulted in each operation, it is also equal to the number of pages faulted.

Counter Name	Counter Type	Counter Description
Cache Faults/sec	RateOfCountsPerSecond32	Cache Faults/sec is the rate at which faults occur when a page sought in the file system cache is not found and must be retrieved from elsewhere in memory (a soft fault) or from disk (a hard fault). The file system cache is an area of physical memory that stores recently used pages of data for applications. Cache activity is a reliable indicator of most application I/O operations. This counter shows the number of faults, without regard for the number of pages faulted in each operation.
Commit Limit	NumberOfItems64	Commit Limit is the amount of virtual memory that can be committed without having to extend the paging file(s). It is measured in bytes. Committed memory is the physical memory which has space reserved on the disk paging files. There can be one paging file on each logical drive). If the paging file(s) are be expanded, this limit increases accordingly. This counter displays the last observed value only; it is not an average.
Page Faults/sec	RateOfCountsPerSecond32	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.
Available Bytes	NumberOfItems64	Available Bytes is the amount of physical memory, in bytes, available to processes running on the computer. It is calculated by adding the amount of space on the Zeroed, Free, and Standby memory lists. Free memory is ready for use; Zeroed memory consists of pages of memory filled with zeros to prevent subsequent processes from seeing data used by a previous process; Standby memory is memory that has been removed from a process' working set (its physical memory) on route to disk, but is still available to be recalled. This counter displays the last observed value only; it is not an average.
Committed Bytes	NumberOfItems64	Committed Bytes is the amount of committed virtual memory, in bytes. Committed memory is the physical memory which has space reserved on the disk paging file(s). 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 Output/sec	RateOfCountsPerSecond32	Pages Output/sec is the rate at which pages are written to disk to free up space in physical memory. Pages are written

Counter Name	Counter Type	Counter Description
		back to disk only if they are changed in physical memory, so they are likely to hold data, not code. A high rate of pages output might indicate a memory shortage. Windows writes more pages back to disk to free up space when physical memory is in short supply. This counter shows the number of pages, and can be compared to other counts of pages, without conversion.
Pool Paged Bytes	NumberOfItems64	Pool Paged Bytes is the size, in bytes, of the paged pool, an area of system memory (physical memory used by the operating system) for objects that can be written to disk when they are not being used. Memory\Pool Paged Bytes is calculated differently than Process\Pool Paged Bytes, so it might not equal Process\Pool Paged Bytes_Total. This counter displays the last observed value only; it is not an average.
Pool Nonpaged Bytes	NumberOfItems64	Pool Nonpaged Bytes is the size, in bytes, of the nonpaged pool, an area of system memory (physical memory used by the operating system) for objects that cannot be written to disk, but must remain in physical memory as long as they are allocated. Memory\Pool Nonpaged Bytes is calculated differently than Process\Pool Nonpaged Bytes, so it might not equal Process\Pool Nonpaged Bytes_Total. This counter displays the last observed value only; it is not an average.
Page Reads/sec	RateOfCountsPerSecond32	Page Reads/sec is the rate at which the disk was read to resolve hard page faults. It shows the number of reads operations, without regard to the number of pages retrieved in each operation. Hard page faults occur when a process references a page in virtual memory that is not in working set or elsewhere in physical memory, and must be retrieved from disk. This counter is a primary indicator of the kinds of faults that cause system-wide delays. It includes read operations to satisfy faults in the file system cache (usually requested by applications) and in non-cached mapped memory files. Compare the value of Memory\Pages Reads/sec to the value of Memory\Pages Input/sec to determine the average number of pages read during each operation.
Demand Zero Faults/sec	RateOfCountsPerSecond32	Demand Zero Faults/sec is the rate at which a zeroed page is required to satisfy the fault. Zeroed pages, pages emptied of previously stored data and filled with zeros, are a security feature of Windows that prevent processes from seeing data stored by earlier processes that used the memory space. Windows maintains a list of zeroed pages to accelerate this process. This counter shows the number of faults, without regard to the number of pages retrieved to satisfy the fault. This counter displays the difference between the values observed in the last two samples, divided by the duration of

Counter Name	Counter Type	Counter Description
		the sample interval.
Pages/sec	RateOfCountsPerSecond32	Pages/sec is the rate at which pages are read from or written to disk to resolve hard page faults. This counter is a primary indicator of the kinds of faults that cause system-wide delays. It is the sum of Memory\Pages Input/sec and Memory\Pages Output/sec. It is counted in numbers of pages, so it can be compared to other counts of pages, such as Memory\Page Faults/sec, without conversion. It includes pages retrieved to satisfy faults in the file system cache (usually requested by applications) non-cached mapped memory files.
Pages Input/sec	RateOfCountsPerSecond32	Pages Input/sec is the rate at which pages are read from disk to resolve hard page faults. Hard page faults occur when a process refers to a page in virtual memory that is not in its working set or elsewhere in physical memory, and must be retrieved from disk. When a page is faulted, the system tries to read multiple contiguous pages into memory to maximize the benefit of the read operation. Compare the value of Memory\Pages Input/sec to the value of Memory\Page Reads/sec to determine the average number of pages read into memory during each read operation.

Category: MSDTC Bridge 3.0.0.0

CategoryType: SingleInstance.

MSDTC Bridge 3.0.0.0 performance counters

Counter Name	Counter Type	Counter Description
Average participant prepare response time	AverageTimer32	Average time in milliseconds for the WS-AT service to receive a Prepare message response from a participant.
Faults sent count/sec	RateOfCountsPerSecond32	The number of Fault messages that the WS-AT service has sent per second.
Average participant prepare response time Base	AverageBase	Base counter for the 'Average participant prepare response time' counter.
Average participant commit response time Base	AverageBase	Base counter for the 'Average participant commit response time' counter.
Average participant	AverageTimer32	Average time in milliseconds for the WS-AT

Counter Name	Counter Type	Counter Description
commit response time		service to receive a Commit message response from a participant.
Faults received count/sec	RateOfCountsPerSecond32	The number of Fault messages that the WS-AT service has received per second.
Prepare retry count/sec	RateOfCountsPerSecond32	The number of Prepare retry messages that the WS-AT service has sent per second.
Message send failures/sec	RateOfCountsPerSecond32	The number of WS-AT protocol messages that the WS-AT service failed to send per second.
Commit retry count/sec	RateOfCountsPerSecond32	The number of Commit retry messages that the WS-AT service has sent per second.
Replay retry count/sec	RateOfCountsPerSecond32	The number of Replay retry messages that the WS-AT service has sent per second.
Prepared retry count/sec	RateOfCountsPerSecond32	The number of Prepared retry messages that the WS-AT service has sent per second.

Category: MSDTC Bridge 4.0.0.0

CategoryType: SingleInstance.

MSDTC Bridge 4.0.0.0 performance counters

Counter Name	Counter Type	Counter Description
Average participant prepare response time	AverageTimer32	Average time in milliseconds for the WS-AT service to receive a Prepare message response from a participant.
Faults sent count/sec	RateOfCountsPerSecond32	The number of Fault messages that the WS-AT service has sent per second.
Average participant prepare response time Base	AverageBase	Base counter for the 'Average participant prepare response time' counter.
Average participant commit response time Base	AverageBase	Base counter for the 'Average participant commit response time' counter.
Average participant commit response time	AverageTimer32	Average time in milliseconds for the WS-AT service to receive a Commit message response

Counter Name	Counter Type	Counter Description
		from a participant.
Faults received count/sec	RateOfCountsPerSecond32	The number of Fault messages that the WS-AT service has received per second.
Prepare retry count/sec	RateOfCountsPerSecond32	The number of Prepare retry messages that the WS-AT service has sent per second.
Message send failures/sec	RateOfCountsPerSecond32	The number of WS-AT protocol messages that the WS-AT service failed to send per second.
Commit retry count/sec	RateOfCountsPerSecond32	The number of Commit retry messages that the WS-AT service has sent per second.
Replay retry count/sec	RateOfCountsPerSecond32	The number of Replay retry messages that the WS-AT service has sent per second.
Prepared retry count/sec	RateOfCountsPerSecond32	The number of Prepared retry messages that the WS-AT service has sent per second.

Category: NBT Connection

CategoryType: MultiInstance.

The NBT Connection performance object consists of counters that measure the rates at which bytes are sent and received over the NBT connection between the local computer and a remote computer. The connection is identified by the name of the remote computer.

Instance: Total

Counter Name	Counter Type	Counter Description
Bytes Received/sec	RateOfCountsPerSecond64	Bytes Received/sec is the rate at which bytes are received by the local computer over an NBT connection to some remote computer. All the bytes received by the local computer over the particular NBT connection are counted.
Bytes Sent/sec	RateOfCountsPerSecond64	Bytes Sent/sec is the rate at which bytes are sent by the local computer over an NBT connection to some remote computer. All the bytes sent by the local computer over the particular NBT connection are counted.
Bytes Total/sec	RateOfCountsPerSecond64	Bytes Total/sec is the rate at which bytes are sent or received by the local computer over an NBT connection to some remote

Counter Name	Counter Type	Counter Description
		computer. All the bytes sent or received by the local computer over the particular NBT connection are counted.

Category: Network Interface

CategoryType: MultiInstance.

The Network Interface performance object consists of counters that measure the rates at which bytes and packets are sent and received over a TCP/IP network connection. It includes counters that monitor connection errors.

Instance: VMware Accelerated AMD PCNet Adapter - Paketplaner-Miniport

Counter Name	Counter Type	Counter Description
Bytes Total/sec	RateOfCountsPerSecond64	Bytes Total/sec is the rate at which bytes are sent and received over each network adapter, including framing characters. Network Interface\Bytes Received/sec is a sum of Network Interface\Bytes Received/sec and Network Interface\Bytes Sent/sec.
Packets/sec	RateOfCountsPerSecond32	Packets/sec is the rate at which packets are sent and received on the network interface.
Packets Received/sec	RateOfCountsPerSecond32	Packets Received/sec is the rate at which packets are received on the network interface.
Packets Sent/sec	RateOfCountsPerSecond32	Packets Sent/sec is the rate at which packets are sent on the network interface.
Current Bandwidth	NumberOfItems32	Current Bandwidth is an estimate of the current bandwidth of the network interface in bits per second (BPS). For interfaces that do not vary in bandwidth or for those where no accurate estimation can be made, this value is the nominal bandwidth.
Bytes Received/sec	RateOfCountsPerSecond32	Bytes Received/sec is the rate at which bytes are received over each network adapter, including framing characters. Network Interface\Bytes Received/sec is a subset of Network Interface\Bytes Total/sec.

Counter Name	Counter Type	Counter Description
Packets Received Unicast/sec	RateOfCountsPerSecond32	Packets Received Unicast/sec is the rate at which (subnet) unicast packets are delivered to a higher-layer protocol.
Packets Received Non-Unicast/sec	RateOfCountsPerSecond32	Packets Received Non-Unicast/sec is the rate at which non-unicast (subnet broadcast or subnet multicast) packets are delivered to a higher-layer protocol.
Packets Received Discarded	NumberOfItems32	Packets Received Discarded is the number of inbound packets that were chosen to be discarded even though no errors had been detected to prevent their delivery to a higher-layer protocol. One possible reason for discarding packets could be to free up buffer space.
Packets Received Errors	NumberOfItems32	Packets Received Errors is the number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol.
Packets Received Unknown	NumberOfItems32	Packets Received Unknown is the number of packets received through the interface that were discarded because of an unknown or unsupported protocol.
Bytes Sent/sec	RateOfCountsPerSecond32	Bytes Sent/sec is the rate at which bytes are sent over each network adapter, including framing characters. Network Interface\Bytes Sent/sec is a subset of Network Interface\Bytes Total/sec.
Packets Sent Unicast/sec	RateOfCountsPerSecond32	Packets Sent Unicast/sec is the rate at which packets are requested to be transmitted to subnet-unicast addresses by higher-level protocols. The rate includes the packets that were discarded or not sent.
Packets Sent Non-Unicast/sec	RateOfCountsPerSecond32	Packets Sent Non-Unicast/sec is the rate at which packets are requested to be transmitted to non-unicast (subnet broadcast or subnet multicast) addresses by higher-level protocols. The rate includes the packets that were discarded or not sent.
Packets Outbound Discarded	NumberOfItems32	Packets Outbound Discarded is the number of outbound packets that were chosen to be discarded even though no errors had been detected to prevent transmission. One possible reason for discarding packets could be to free up buffer space.
Packets Outbound Errors	NumberOfItems32	Packets Outbound Errors is the number of outbound packets that could not be transmitted because of errors.
Output Queue	NumberOfItems32	Output Queue Length is the length of the output packet

Counter Name	Counter Type	Counter Description
Length		queue (in packets). If this is longer than two, there are delays and the bottleneck should be found and eliminated, if possible. Since the requests are queued by the Network Driver Interface Specification (NDIS) in this implementation, this will always be 0.

Instance: MS TCP Loopback interface

Instance: Ethernetadapter der AMD-PCNET-Familie - Paketplaner-Miniport

Category: Objects

CategoryType: SingleInstance.

The Object performance object consists of counters that monitor logical objects in the system, such as processes, threads, mutexes, and semaphores. This information can be used to detect the unnecessary consumption of computer resources. Each object requires memory to store basic information about the object.

Counter Name	Counter Type	Counter Description
Semaphores	NumberOfItems32	Semaphores is the number of semaphores in the computer at the time of data collection. This is an instantaneous count, not an average over the time interval. Threads use semaphores to obtain exclusive access to data structures that they share with other threads.
Mutexes	NumberOfItems32	Mutexes counts the number of mutexes in the computer at the time of data collection. This is an instantaneous count, not an average over the time interval. Mutexes are used by threads to assure only one thread is executing a particular section of code.
Sections	NumberOfItems32	Sections is the number of sections in the computer at the time of data collection. This is an instantaneous count, not an average over the time interval. A section is a portion of virtual memory created by a process for storing data. A process can share sections with other processes.
Processes	NumberOfItems32	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.
Threads	NumberOfItems32	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 execute instructions in a processor.
Events	NumberOfItems32	Events is the number of events in the computer at the time of data collection. This is an instantaneous count, not an average over the time

Counter Name	Counter Type	Counter Description
	2	interval. An event is used when two or more threads try to synchronize execution.

Category: Paging File

CategoryType: MultiInstance.

The Paging File performance object consists of counters that monitor the paging file(s) on the computer. The paging file is a reserved space on disk that backs up committed physical memory on the computer.

Instance: _Total

Counter Name	Counter Type	Counter Description
% Usage	RawFraction	The amount of the Page File instance in use in percent. See also Process\Page File Bytes.
% Usage Peak	RawFraction	The peak usage of the Page File instance in percent. See also Process\Page File Bytes Peak.

Instance: \??\C:\pagefile.sys

Category: PhysicalDisk

CategoryType: MultiInstance.

The Physical Disk performance object consists of counters that monitor hard or fixed disk drive on a computer. Disks are used to store file, program, and paging data and are read to retrieve these items, and written to record changes to them. The values of physical disk counters are sums of the values of the logical disks (or partitions) into which they are divided.

Instance: _Total

Counter Name	Counter Type	Counter Description
Current Disk Queue Length	NumberOfItems32	Current Disk Queue Length is the number of requests outstanding on the disk at the time the performance data is collected. It also includes requests in service at the time of the collection. This is a instantaneous snapshot, not an average over the time interval. Multi-spindle disk devices can have multiple requests that are active at one time, but other concurrent

Counter Name	Counter Type	Counter Description
		requests are awaiting service. This counter might reflect a transitory high or low queue length, but if there is a sustained load on the disk drive, it is likely that this will be consistently high. Requests experience delays proportional to the length of this queue minus the number of spindles on the disks. For good performance, this difference should average less than two.
% Disk Time	542573824	% Disk Time is the percentage of elapsed time that the selected disk drive was busy servicing read or write requests.
Avg. Disk Queue Length	5571840	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.
% Disk Read Time	542573824	% Disk Read Time is the percentage of elapsed time that the selected disk drive was busy servicing read requests.
Avg. Disk Read Queue Length	5571840	Avg. Disk Read Queue Length is the average number of read requests that were queued for the selected disk during the sample interval.
% Disk Write Time	542573824	% Disk Write Time is the percentage of elapsed time that the selected disk drive was busy servicing write requests.
Avg. Disk Write Queue Length	5571840	Avg. Disk Write Queue Length is the average number of write requests that were queued for the selected disk during the sample interval.
Avg. Disk sec/Transfer	AverageTimer32	Avg. Disk sec/Transfer is the time, in seconds, of the average disk transfer.
Avg. Disk sec/Read	AverageTimer32	Avg. Disk sec/Read is the average time, in seconds, of a read of data from the disk.
Avg. Disk sec/Write	AverageTimer32	Avg. Disk sec/Write is the average time, in seconds, of a write of data to the disk.
Disk Transfers/sec	RateOfCountsPerSecond32	Disk Transfers/sec is the rate of read and write operations on the disk.

Counter Name	Counter Type	Counter Description
Disk Reads/sec	RateOfCountsPerSecond3 2	Disk Reads/sec is the rate of read operations on the disk.
Disk Writes/sec	RateOfCountsPerSecond3 2	Disk Writes/sec is the rate of write operations on the disk.
Disk Bytes/sec	RateOfCountsPerSecond6 4	Disk Bytes/sec is the rate bytes are transferred to or from the disk during write or read operations.
Disk Read Bytes/sec	RateOfCountsPerSecond6 4	Disk Read Bytes/sec is the rate at which bytes are transferred from the disk during read operations.
Disk Write Bytes/sec	RateOfCountsPerSecond6 4	Disk Write Bytes/sec is rate at which bytes are transferred to the disk during write operations.
Avg. Disk Bytes/Transfer	AverageCount64	Avg. Disk Bytes/Transfer is the average number of bytes transferred to or from the disk during write or read operations.
Avg. Disk Bytes/Read	AverageCount64	Avg. Disk Bytes/Read is the average number of bytes transferred from the disk during read operations.
Avg. Disk Bytes/Write	AverageCount64	Avg. Disk Bytes/Write is the average number of bytes transferred to the disk during write operations.
% Idle Time	542573824	% Idle Time reports the percentage of time during the sample interval that the disk was idle.
Split IO/Sec	RateOfCountsPerSecond3 2	Split IO/Sec reports the rate at which I/Os to the disk were split into multiple I/Os. A split I/O may result from requesting data of a size that is too large to fit into a single I/O or that the disk is fragmented.

Instance: 0 C:

Instance: 2 D:

Instance: 1 D:

Category: Print Queue

CategoryType: MultiInstance.

Displays performance statistics about a Print Queue.

Instance: _Total

Counter Name	Counter Type	Counter Description
Total Jobs Printed	NumberOfItems32	Total number of jobs printed on a print queue since the last restart.
Bytes Printed/sec	RateOfCountsPerSecond64	Number of bytes per second printed on a print queue.
Total Pages Printed	NumberOfItems32	Total number of pages printed through GDI on a print queue since the last restart.
Jobs	NumberOfItems32	Current number of jobs in a print queue.
References	NumberOfItems32	Current number of references (open handles) to this printer.
Max References	NumberOfItems32	Peak number of references (open handles) to this printer.
Jobs Spooling	NumberOfItems32	Current number of spooling jobs in a print queue.
Max Jobs Spooling	NumberOfItems32	Maximum number of spooling jobs in a print queue since last restart.
Out of Paper Errors	NumberOfItems32	Total number of out of paper errors in a print queue since the last restart.
Not Ready Errors	NumberOfItems32	Total number of printer not ready errors in a print queue since the last restart.
Job Errors	NumberOfItems32	Total number of job errors in a print queue since last restart.
Enumerate Network Printer Calls	NumberOfItems32	Total number of calls from browse clients to this print server to request network browse lists since last restart.
Add Network Printer Calls	NumberOfItems32	Total number of calls from other print servers to add shared network printers to this server since last restart.

Instance: Microsoft XPS Document Writer

Category: Process

CategoryType: MultiInstance.

The Process performance object consists of counters that monitor running application program and system processes. All the threads in a process share the same address space and have access to the same data.

Instance: msseces#1

Counter Name	Counter Type	Counter Description
% Processor Time	Timer100Ns	% Processor Time is the percentage of elapsed time that all of process threads used the processor to execution instructions. An instruction is the basic unit of execution in a computer, a thread is the object that executes instructions, and a process is the object created when a program is run. Code executed to handle some hardware interrupts and trap conditions are included in this count.
% User Time	Timer100Ns	% User Time is the percentage of elapsed time that the process threads spent executing code in user mode. Applications, environment subsystems, and integral subsystems execute in user mode. Code executing in user mode cannot damage the integrity of the Windows executive, kernel, and device drivers. Unlike some early operating systems, Windows uses process boundaries for subsystem protection in addition to the traditional protection of user and privileged modes. Some work done by Windows on behalf of the application might appear in other subsystem processes in addition to the privileged time in the process.
% Privileged Time	Timer100Ns	% Privileged Time is the percentage of elapsed time that the process threads spent executing code in privileged mode. When a Windows system service is called, the service will often run in privileged mode to gain access to system-private data. Such data is protected from access by threads executing in user mode. Calls to the system can be explicit or implicit, such as page faults or interrupts. Unlike some early operating systems, Windows uses process boundaries for subsystem protection in addition to the traditional protection of user and privileged modes. Some work done by Windows on behalf of the application might appear in other subsystem processes in addition to the privileged time in the process.
Virtual Bytes Peak	NumberOfItems64	Virtual Bytes Peak is the maximum size, in bytes, of virtual address space the process has used at any one time. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. However, virtual space is finite, and the process might limit its ability to load libraries.
Virtual Bytes	NumberOfItems64	Virtual Bytes is the current size, in bytes, of the virtual address space the process is using. Use of virtual address space does not necessarily imply corresponding use of either

Counter Name	Counter Type	Counter Description
		disk or main memory pages. Virtual space is finite, and the process can limit its ability to load libraries.
Page Faults/sec	RateOfCountsPerSecond32	Page Faults/sec is the rate at which page faults by the threads executing in this process are occurring. A page fault occurs when a thread refers to a virtual memory page that is not in its working set in main memory. This may not cause the page to be fetched from disk if it is on the standby list and hence already in main memory, or if it is in use by another process with whom the page is shared.
Working Set Peak	NumberOfItems64	Working Set Peak is the maximum size, in bytes, of the Working Set of this process at any point in time. The Working Set is the set of memory pages touched recently by the threads in the process. If free memory in the computer is above a threshold, pages are left in the Working Set of a process even if they are not in use. When free memory falls below a threshold, pages are trimmed from Working Sets. If they are needed they will then be soft-faulted back into the Working Set before they leave main memory.
Working Set	NumberOfItems64	Working Set is the current size, in bytes, of the Working Set of this process. The Working Set is the set of memory pages touched recently by the threads in the process. If free memory in the computer is above a threshold, pages are left in the Working Set of a process even if they are not in use. When free memory falls below a threshold, pages are trimmed from Working Sets. If they are needed they will then be soft-faulted back into the Working Set before leaving main memory.
Page File Bytes Peak	NumberOfItems64	Page File Bytes Peak is the maximum number of bytes this process has used in the paging file(s). Paging files are used to store pages of memory used by the process that are not contained in other files. Paging files are shared by all processes, and the lack of space in paging files can prevent other processes from allocating memory.
Page File Bytes	NumberOfItems64	Page File Bytes is the current number of bytes that this process has used in the paging file(s). Paging files are used to store pages of memory used by the process that are not contained in other files. Paging files are shared by all processes, and the lack of space in paging files can prevent other processes from allocating memory.
Private Bytes	NumberOfItems64	Private Bytes is the current size, in bytes, of memory that this process has allocated that cannot be shared with other processes.

Counter Name	Counter Type	Counter Description
Thread Count	NumberOfItems32	The number of threads currently active in this process. An instruction is the basic unit of execution in a processor, and a thread is the object that executes instructions. Every running process has at least one thread.
Priority Base	NumberOfItems32	The current base priority of this process. Threads within a process can raise and lower their own base priority relative to the process' base priority.
Elapsed Time	ElapsedTime	The total elapsed time, in seconds, that this process has been running.
ID Process	NumberOfItems32	ID Process is the unique identifier of this process. ID Process numbers are reused, so they only identify a process for the lifetime of that process.
Creating Process ID	NumberOfItems32	The Creating Process ID value is the Process ID of the process that created the process. The creating process may have terminated, so this value may no longer identify a running process.
Pool Paged Bytes	NumberOfItems32	Pool Paged Bytes is the size, in bytes, of the paged pool, an area of system memory (physical memory used by the operating system) for objects that can be written to disk when they are not being used. Memory\Pool Paged Bytes is calculated differently than Process\Pool Paged Bytes, so it might not equal Process\Pool Paged Bytes_Total. This counter displays the last observed value only; it is not an average.
Pool Nonpaged Bytes	NumberOfItems32	Pool Nonpaged Bytes is the size, in bytes, of the nonpaged pool, an area of system memory (physical memory used by the operating system) for objects that cannot be written to disk, but must remain in physical memory as long as they are allocated. Memory\Pool Nonpaged Bytes is calculated differently than Process\Pool Nonpaged Bytes, so it might not equal Process\Pool Nonpaged Bytes_Total. This counter displays the last observed value only; it is not an average.
Handle Count	NumberOfItems32	The total number 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.
IO Read Operations/sec	RateOfCountsPerSecond64	The rate at which the process is issuing read I/O operations. This counter counts all I/O activity generated by the process to include file, network and device I/Os.

Counter Name	Counter Type	Counter Description
IO Write Operations/sec	RateOfCountsPerSecond64	The rate at which the process is issuing write I/O operations. This counter counts all I/O activity generated by the process to include file, network and device I/Os.
IO Data Operations/sec	RateOfCountsPerSecond64	The rate at which the process is issuing read and write I/O operations. This counter counts all I/O activity generated by the process to include file, network and device I/Os.
IO Other Operations/sec	RateOfCountsPerSecond64	The rate at which the process is issuing I/O operations that are neither read nor write operations (for example, a control function). This counter counts all I/O activity generated by the process to include file, network and device I/Os.
IO Read Bytes/sec	RateOfCountsPerSecond64	The rate at which the process is reading bytes from I/O operations. This counter counts all I/O activity generated by the process to include file, network and device I/Os.
IO Write Bytes/sec	RateOfCountsPerSecond64	The rate at which the process is writing bytes to I/O operations. This counter counts all I/O activity generated by the process to include file, network and device I/Os.
IO Data Bytes/sec	RateOfCountsPerSecond64	The rate at which the process is reading and writing bytes in I/O operations. This counter counts all I/O activity generated by the process to include file, network and device I/Os.
IO Other Bytes/sec	RateOfCountsPerSecond64	The rate at which the process is issuing bytes to I/O operations that do not involve data such as control operations. This counter counts all I/O activity generated by the process to include file, network and device I/Os.

Instance: wmioprse

Instance: Idle

... and many more.

Category: Processor

CategoryType: MultiInstance.

The Processor performance object consists of counters that measure aspects of processor activity. The processor is the part of the computer that performs arithmetic and logical computations, initiates operations on peripherals, and runs the threads of processes. A computer can have multiple processors. The processor object represents each processor as an instance of the object.

Instance: _Total

Counter Name	Counter Type	Counter Description
% Processor Time	Timer100NsInverse	% Processor Time is the percentage of elapsed time that the processor spends to execute a non-Idle thread. It is calculated by measuring the duration of 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%.
% User Time	Timer100Ns	% User Time is the percentage of elapsed time the processor spends in the user mode. User mode is a restricted processing mode designed for applications, environment subsystems, and integral subsystems. The alternative, privileged mode, is designed for operating system components and allows direct access to hardware and all memory. The operating system switches application threads to privileged mode to access operating system services. This counter displays the average busy time as a percentage of the sample time.
% Privileged Time	Timer100Ns	% Privileged Time is the percentage of elapsed time that the process threads spent executing code in privileged mode. When a Windows system service is called, the service will often run in privileged mode to gain access to system-private data. Such data is protected from access by threads executing in user mode. Calls to the system can be explicit or implicit, such as page faults or interrupts. Unlike some early operating systems, Windows uses process boundaries for subsystem protection in addition to the traditional protection of user and privileged modes. Some work done by Windows on behalf of the application might appear in other subsystem processes in addition to the privileged time in the process.
Interrupts/sec	RateOfCountsPerSecond32	Interrupts/sec is the average rate, in incidents per second, at which the processor received and serviced hardware interrupts. It does not include deferred procedure calls (DPCs), which are counted separately. This value is an indirect indicator of the activity of devices that generate interrupts, such as the system clock, the mouse, disk drivers, data communication lines, network interface cards, and other peripheral devices. These devices normally interrupt the processor when they have completed a task or require attention. Normal thread execution is suspended. The system clock typically interrupts the processor every 10 milliseconds, creating a background of interrupt activity. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Counter Name	Counter Type	Counter Description
% DPC Time	Timer100Ns	% DPC Time is the percentage of time that the processor spent receiving and servicing deferred procedure calls (DPCs) during the sample interval. DPCs are interrupts that run at a lower priority than standard interrupts. % DPC Time is a component of % Privileged Time because DPCs are executed in privileged mode. They are counted separately and are not a component of the interrupt counters. This counter displays the average busy time as a percentage of the sample time.
% Interrupt Time	Timer100Ns	% Interrupt Time is the time the processor spends receiving and servicing hardware interrupts during sample intervals. This value is an indirect indicator of the activity of devices that generate interrupts, such as the system clock, the mouse, disk drivers, data communication lines, network interface cards and other peripheral devices. These devices normally interrupt the processor when they have completed a task or require attention. Normal thread execution is suspended during interrupts. Most system clocks interrupt the processor every 10 milliseconds, creating a background of interrupt activity. suspends normal thread execution during interrupts. This counter displays the average busy time as a percentage of the sample time.
DPCs Queued/sec	RateOfCountsPerSecond32	DPCs Queued/sec is the average rate, in incidents per second, at which deferred procedure calls (DPCs) were added to the processor's DPC queue. DPCs are interrupts that run at a lower priority than standard interrupts. Each processor has its own DPC queue. This counter measures the rate that DPCs are added to the queue, not the number of DPCs in the queue. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.
DPC Rate	NumberOfItems32	DPC Rate is the rate at which deferred procedure calls (DPCs) were added to the processors DPC queues between the timer ticks of the processor clock. DPCs are interrupts that run at a lower priority than standard interrupts. Each processor has its own DPC queue. This counter measures the rate that DPCs were added to the queue, not the number of DPCs in the queue. This counter displays the last observed value only; it is not an average.
% Idle Time	Timer100Ns	% Idle Time is the percentage of time the processor is idle during the sample interval
% C1 Time	Timer100Ns	% C1 Time is the percentage of time the processor spends in the C1 low-power idle state. % C1 Time is a subset of the total processor idle time. C1 low-power idle state enables the processor to maintain its entire context and quickly return to

Counter Name	Counter Type	Counter Description
		the running state. Not all systems support the % C1 state.
% C2 Time	Timer100Ns	% C2 Time is the percentage of time the processor spends in the C2 low-power idle state. % C2 Time is a subset of the total processor idle time. C2 low-power idle state enables the processor to maintain the context of the system caches. The C2 power state is a lower power and higher exit latency state than C1. Not all systems support the C2 state.
% C3 Time	Timer100Ns	% C3 Time is the percentage of time the processor spends in the C3 low-power idle state. % C3 Time is a subset of the total processor idle time. When the processor is in the C3 low-power idle state it is unable to maintain the coherency of its caches. The C3 power state is a lower power and higher exit latency state than C2. Not all systems support the C3 state.
C1 Transitions/sec	RateOfCountsPerSecond64	C1 Transitions/sec is the rate that the CPU enters the C1 low-power idle state. The CPU enters the C1 state when it is sufficiently idle and exits this state on any interrupt. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.
C2 Transitions/sec	RateOfCountsPerSecond64	C2 Transitions/sec is the rate that the CPU enters the C2 low-power idle state. The CPU enters the C2 state when it is sufficiently idle and exits this state on any interrupt. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.
C3 Transitions/sec	RateOfCountsPerSecond64	C3 Transitions/sec is the rate that the CPU enters the C3 low-power idle state. The CPU enters the C3 state when it is sufficiently idle and exits this state on any interrupt. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Instance: 0

Instance: 1

Category: PSched Flow

CategoryType: MultiInstance.

Flow statistics from the packet scheduler

Category: PSched Pipe

CategoryType: MultiInstance.

Pipe statistics from the packet scheduler

Instance: VMware Accelerated AMD PCNet Adapter - Paketplaner-Miniport

Counter Name	Counter Type	Counter Description
Out of packets	NumberOfItems32	The number of times PSched has been unable to allocate a packet
Flows opened	NumberOfItems32	The number of flows opened on this pipe (some of which may now be closed)
Flows closed	NumberOfItems32	The number of flows that have been closed
Flows rejected	NumberOfItems32	The number of flow creations that were rejected
Flows modified	NumberOfItems32	The of times a flow has been modified
Flow mods rejected	NumberOfItems32	The number of times a flow modification has been rejected
Max simultaneous flows	NumberOfItems32	The maximum number of flows that have been simultaneously open on this pipe
Nonconforming packets scheduled	NumberOfItems32	The number of packets that have entered the packet scheduler at a rate which exceeded that packet's flow parameters
Nonconforming packets scheduled/sec	RateOfCountsPerSecond32	The rate at which nonconforming packets have entered the packet scheduler
Average packets in shaper	NumberOfItems32	The average number of packets in the shaper over the last sampling period
Max packets in shaper	NumberOfItems32	The maximum number of packets that have ever simultaneously been in the shaper
Average packets in sequencer	NumberOfItems32	The average number of packets in the sequencer over the last sampling period
Max packets in sequencer	NumberOfItems32	The maximum number of packets that have ever

Counter Name	Counter Type	Counter Description
		simultaneously been in the sequencer
Max packets in netcard	NumberOfItems32	The maximum number of packets ever simultaneously in the network card
Average packets in netcard	NumberOfItems32	The average number of packets in the network card over the last sampling period
Nonconforming packets transmitted	NumberOfItems32	The number of packets that have been sent by the packet scheduler at a rate which exceeded that packet's flow parameters
Nonconforming packets transmitted/sec	RateOfCountsPerSecond32	The rate at which nonconforming packets have been sent by the packet scheduler

Instance: Ethernetadapter der AMD-PCNET-Familie - Paketplaner-Miniport

Category: Redirector

CategoryType: SingleInstance.

The Redirector performance object consists of counter that monitor network connections originating at the local computer.

Counter Name	Counter Type	Counter Description
Reads Denied/sec	RateOfCountsPerSecond32	Reads Denied/sec is the rate at which the server is unable to accommodate requests for Raw Reads. When a read is much larger than the server's negotiated buffer size, the Redirector requests a Raw Read which, if granted, would permit the transfer of the data without lots of protocol overhead on each packet. To accomplish this the server must lock out other requests, so the request is denied if the server is really busy.
Write Packets Small/sec	RateOfCountsPerSecond32	Write Packets Small/sec is the rate at which writes are made by applications that are less than one-fourth of the server's negotiated buffer size. Too many of these could indicate a waste of buffers on the server. This counter is incremented once for each write: it counts writes, not packets.
Network Errors/sec	RateOfCountsPerSecond32	Network Errors/sec is the rate at which serious unexpected errors are occurring. Such errors generally indicate that the Redirector and one or more Servers are having serious communication difficulties. For example an SMB (Server Manager Block) protocol error is a Network Error. An entry is written to the System Event Log and provide details.

Counter Name	Counter Type	Counter Description
Writes Denied/sec	RateOfCountsPerSecond32	Writes Denied/sec is the rate at which the server is unable to accommodate requests for Raw Writes. When a write is much larger than the server's negotiated buffer size, the Redirector requests a Raw Write which, if granted, would permit the transfer of the data without lots of protocol overhead on each packet. To accomplish this the server must lock out other requests, so the request is denied if the server is really busy.
Writes Large/sec	RateOfCountsPerSecond32	Writes Large/sec is the rate at which writes are made by applications that are over 2 times the server's negotiated buffer size. Too many of these could place a strain on server resources. This counter is incremented once for each write: it counts writes, not packets.
File Write Operations/sec	RateOfCountsPerSecond32	File Write Operations/sec is the rate at which applications are sending data to the Redirector. Each call to a file system or similar Application Program Interface (API) call counts as one operation.
Read Packets Small/sec	RateOfCountsPerSecond32	Read Packets Small/sec is the rate at which reads less than one-fourth of the server's negotiated buffer size are made by applications. Too many of these could indicate a waste of buffers on the server. This counter is incremented once for each read. It does not count packets.
Write Packets/sec	RateOfCountsPerSecond32	Write Packets/sec is the rate at which writes are being sent to the network. Each time a single packet is sent with a request to write remote data, this counter is incremented by one.
Write Operations Random/sec	RateOfCountsPerSecond32	Write Operations Random/sec is the rate at which, on a file-by-file basis, writes are made that are not sequential. If a write is made using a particular file handle, and then is followed by another write that is not immediately the next contiguous byte, this counter is incremented by one.
Server Disconnects	NumberOfItems32	Server Disconnects counts the number of times a Server has disconnected your Redirector. See also Server Reconnects.
Connects Windows NT	NumberOfItems32	Connects Windows NT counts the connections to Windows 2000 or earlier computers.
Current Commands	NumberOfItems32	Current Commands counts the number of requests to the Redirector that are currently queued for service. If this number is much larger than the number of network adapter cards installed in the computer, then the network(s) and/or the server(s) being accessed are seriously bottlenecked.

Counter Name	Counter Type	Counter Description
Server Sessions Hung	NumberOfItems32	Server Sessions Hung counts the number of active sessions that are timed out and unable to proceed due to a lack of response from the remote server.
Connects Lan Manager 2.1	NumberOfItems32	Connects LAN Manager 2.1 counts connections to LAN Manager 2.1 servers, including LMX servers.
Server Reconnects	NumberOfItems32	Server Reconnects counts the number of times your Redirector has had to reconnect to a server in order to complete a new active request. You can be disconnected by the Server if you remain inactive for too long. Locally even if all your remote files are closed, the Redirector will keep your connections intact for (nominally) ten minutes. Such inactive connections are called Dormant Connections. Reconnecting is expensive in time.
Server Sessions	NumberOfItems32	Server Sessions counts the total number of security objects the Redirector has managed. For example, a logon to a server followed by a network access to the same server will establish one connection, but two sessions.
Connects Lan Manager 2.0	NumberOfItems32	Connects LAN Manager 2.0 counts connections to LAN Manager 2.0 servers, including LMX servers.
Connects Core	NumberOfItems32	Connects Core counts the number of connections you have to servers running the original MS-Net SMB protocol, including MS-Net itself and Xenix and VAX's.
Reads Large/sec	RateOfCountsPerSecond32	Reads Large/sec is the rate at which reads over 2 times the server's negotiated buffer size are made by applications. Too many of these could place a strain on server resources. This counter is incremented once for each read. It does not count packets.
Read Bytes Non-Paging/sec	RateOfCountsPerSecond64	Read Bytes Non-Paging/sec are those bytes read by the Redirector in response to normal file requests by an application when they are redirected to come from another computer. In addition to file requests, this counter includes other methods of reading across the network such as Named Pipes and Transactions. This counter does not count network protocol information, just application data.
Read Bytes Paging/sec	RateOfCountsPerSecond64	Read Bytes Paging/sec is the rate at which the Redirector is attempting to read bytes in response to page faults. Page faults are caused by loading of modules (such as programs and libraries), by a miss in the Cache (see Read Bytes Cache/sec), or by files directly mapped into the address space of applications (a high-performance feature of

Counter Name	Counter Type	Counter Description
		Windows NT).
Read Bytes Network/sec	RateOfCountsPerSecond64	Read Bytes Network/sec is the rate at which applications are reading data across the network. This occurs when data sought in the file system cache is not found there and must be retrieved from the network. Dividing this value by Bytes Received/sec indicates the proportion of application data traveling across the network. (see Bytes Received/sec).
Read Bytes Cache/sec	RateOfCountsPerSecond64	Read Bytes Cache/sec is the rate at which applications are accessing the file system cache by using the Redirector. Some of these data requests are satisfied by retrieving the data from the cache. Requests that miss the Cache cause a page fault (see Read Bytes Paging/sec).
Packets Received/sec	RateOfCountsPerSecond64	Packets Received/sec is the rate at which the Redirector is receiving packets (also called SMBs or Server Message Blocks). Network transmissions are divided into packets. The average number of bytes received in a packet can be obtained by dividing Bytes Received/sec by this counter. Some packets received might not contain incoming data (for example an acknowledgment to a write made by the Redirector would count as an incoming packet).
File Data Operations/sec	RateOfCountsPerSecond32	File Data Operations/sec is the rate at which the Redirector is processing data operations. One operation should include many bytes, since each operation has overhead. The efficiency of this path can be determined by dividing the Bytes/sec by this counter to obtain the average number of bytes transferred per operation.
Bytes Total/sec	RateOfCountsPerSecond64	Bytes Total/sec is the rate the Redirector is processing data bytes. This includes all application and file data in addition to protocol information such as packet headers.
Bytes Received/sec	RateOfCountsPerSecond64	Bytes Received/sec is the rate of bytes coming in to the Redirector from the network. It includes all application data as well as network protocol information (such as packet headers).
Packets/sec	RateOfCountsPerSecond64	Packets/sec is the rate the Redirector is processing data packets. One packet includes (hopefully) many bytes. We say hopefully here because each packet has protocol overhead. You can determine the efficiency of this path by dividing the Bytes/sec by this counter to determine the average number of bytes transferred/packet. You can also divide this counter by Operations/sec to determine the average number of packets per operation, another measure of

Counter Name	Counter Type	Counter Description
		efficiency.
File Read Operations/sec	RateOfCountsPerSecond32	File Read Operations/sec is the rate at which applications are asking the Redirector for data. Each call to a file system or similar Application Program Interface (API) call counts as one operation.
Write Bytes Network/sec	RateOfCountsPerSecond64	Write Bytes Network/sec is the rate at which applications are writing data across the network. This occurs when the file system cache is bypassed, such as for Named Pipes or Transactions, or when the cache writes the bytes to disk to make room for other data. Dividing this counter by Bytes Transmitted/sec will indicate the proportion of application data being to the network (see Transmitted Bytes/sec).
Read Packets/sec	RateOfCountsPerSecond32	Read Packets/sec is the rate at which read packets are being placed on the network. Each time a single packet is sent with a request to read data remotely, this counter is incremented by one.
Read Operations Random/sec	RateOfCountsPerSecond32	Read Operations Random/sec counts the rate at which, on a file-by-file basis, reads are made that are not sequential. If a read is made using a particular file handle, and then is followed by another read that is not immediately the contiguous next byte, this counter is incremented by one.
Write Bytes Cache/sec	RateOfCountsPerSecond64	Write Bytes Cache/sec is the rate at which applications on your computer are writing to the file system cache by using the Redirector. The data might not leave your computer immediately; it can be retained in the cache for further modification before being written to the network. This saves network traffic. Each write of a byte into the cache is counted here.
Packets Transmitted/sec	RateOfCountsPerSecond64	Packets Transmitted/sec is the rate at which the Redirector is sending packets (also called SMBs or Server Message Blocks). Network transmissions are divided into packets. The average number of bytes transmitted in a packet can be obtained by dividing Bytes Transmitted/sec by this counter.
Bytes Transmitted/sec	RateOfCountsPerSecond64	Bytes Transmitted/sec is the rate at which bytes are leaving the Redirector to the network. It includes all application data as well as network protocol information (such as packet headers and the like).
Write Bytes Non-Paging/sec	RateOfCountsPerSecond64	Write Bytes Non-Paging/sec is the rate at which bytes are written by the Redirector in response to normal file outputs by an application when they are redirected to another computer. In addition to file requests, this count includes

Counter Name	Counter Type	Counter Description
		other methods of writing across the network, such as Named Pipes and Transactions. This counter does not count network protocol information, just application data.
Write Bytes Paging/sec	RateOfCountsPerSecond64	Write Bytes Paging/sec is the rate at which the Redirector is attempting to write bytes changed in the pages being used by applications. The program data changed by modules (such as programs and libraries) that were loaded over the network are 'paged out' when no longer needed. Other output pages come from the file system cache (see Write Bytes Cache/sec).

Category: RSVP Interfaces

CategoryType: MultiInstance.

RSVP Interfaces performance counters.

Category: RSVP Service

CategoryType: MultiInstance.

RSVP service performance counters.

Instance: Service

Counter Name	Counter Type	Counter Description
Network Interfaces	NumberOfItems32	The number of local network interfaces visible to, and used by the RSVP service.
Network sockets	NumberOfItems32	The total number of raw sockets opened for the purpose of RSVP signaling.
Timers	NumberOfItems32	The number of timer events scheduled to take place. Shows the activity level of the RSVP service.
RSVP sessions	NumberOfItems32	The current number of active RSVP sessions on the RSVP service.
QoS clients	NumberOfItems32	The number of QoS enabled applications currently active.

Counter Name	Counter Type	Counter Description
QoS-enabled senders	NumberOfItems32	Indicates the number of PATH messages sent for QoS-enabled senders. This number increments each time a PATH is refreshed.
QoS-enabled receivers	NumberOfItems32	Indicates the number of RESV messages sent for QoS-enabled receivers. This number increments each time a RESV is refreshed.
Failed QoS requests	NumberOfItems32	The number of QoS requests generated by QoS-enabled applications that have been rejected by the RSVP service. Failed QoS requests can be caused by invalid QoS requests.
Failed QoS sends	NumberOfItems32	The number of QoS notifications the RSVP service that could not be sent to the QoS applications. Failed QoS sends can be caused by terminated applications.
QoS notifications	NumberOfItems32	The number of QoS notifications delivered to QoS-enabled applications by the QoS RSVP service.
Bytes in QoS notifications	NumberOfItems32	A running total of the number of bytes delivered in QoS Notifications to QoS-enabled applications.

Category: Server

CategoryType: SingleInstance.

The Server performance object consists of counters that measure communication between the local computer and the network.

Counter Name	Counter Type	Counter Description
File Directory Searches	NumberOfItems32	The number of searches for files currently active in the server. Indicates current server activity.
Pool Nonpaged Bytes	NumberOfItems32	The number of bytes of non-pageable computer memory the server is using. This value is useful for determining the values of the MaxNonpagedMemoryUsage value entry in the Windows NT Registry.
Pool Nonpaged Failures	RateOfCountsPerSecond32	The number of times allocations from nonpaged pool have failed. Indicates that the computer's physical memory is too small.
Files Opened Total	NumberOfItems32	The number of successful open attempts performed by the server of behalf of clients. Useful in determining the amount of file I/O, determining overhead for path-based operations,

Counter Name	Counter Type	Counter Description
		and for determining the effectiveness of open locks.
Files Open	NumberOfItems32	The number of files currently opened in the server. Indicates current server activity.
Server Sessions	NumberOfItems32	The number of sessions currently active in the server. Indicates current server activity.
Pool Nonpaged Peak	NumberOfItems32	The maximum number of bytes of nonpaged pool the server has had in use at any one point. Indicates how much physical memory the computer should have.
Context Blocks Queued/sec	RateOfCountsPerSecond32	Context Blocks Queued per second is the rate at which work context blocks had to be placed on the server's FSP queue to await server action.
Logon/sec	RateOfCountsPerSecond32	Logon/sec is the rate of all server logons.
Logon Total	NumberOfItems32	Logon Total includes all interactive logons, network logons, service logons, successful logon, and failed logons since the machine is last rebooted.
Pool Paged Bytes	NumberOfItems32	The number of bytes of pageable computer memory the server is currently using. Can help in determining good values for the MaxPagedMemoryUsage parameter.
Pool Paged Failures	NumberOfItems32	The number of times allocations from paged pool have failed. Indicates that the computer's physical memory or paging file are too small.
Pool Paged Peak	NumberOfItems32	The maximum number of bytes of paged pool the server has had allocated. Indicates the proper sizes of the Page File(s) and physical memory.
Sessions Timed Out	NumberOfItems32	The number of sessions that have been closed due to their idle time exceeding the AutoDisconnect parameter for the server. Shows whether the AutoDisconnect setting is helping to conserve resources.
Sessions Errored Out	NumberOfItems32	The number of sessions that have been closed due to unexpected error conditions or sessions that have reached the autodisconnect timeout and have been disconnected normally.
Sessions Logged Off	NumberOfItems32	The number of sessions that have terminated normally. Useful in interpreting the Sessions Times Out and Sessions

Counter Name	Counter Type	Counter Description
		Errored Out statistics--allows percentage calculations.
Bytes Total/sec	RateOfCountsPerSecond64	The number of bytes the server has sent to and received from the network. This value provides an overall indication of how busy the server is.
Bytes Received/sec	RateOfCountsPerSecond64	The number of bytes the server has received from the network. Indicates how busy the server is.
Bytes Transmitted/sec	RateOfCountsPerSecond64	The number of bytes the server has sent on the network. Indicates how busy the server is.
Sessions Forced Off	NumberOfItems32	The number of sessions that have been forced to logoff. Can indicate how many sessions were forced to logoff due to logon time constraints.
Errors System	NumberOfItems32	The number of times an internal Server Error was detected. Unexpected errors usually indicate a problem with the Server.
Blocking Requests Rejected	RateOfCountsPerSecond32	The number of times the server has rejected blocking SMBs due to insufficient count of free work items. Indicates whether the MaxWorkItem or MinFreeWorkItems server parameters might need to be adjusted.
Work Item Shortages	RateOfCountsPerSecond32	The number of times STATUS_DATA_NOT_ACCEPTED was returned at receive indication time. This occurs when no work item is available or can be allocated to service the incoming request. Indicates whether the InitWorkItems or MaxWorkItems parameters might need to be adjusted.
Errors Logon	NumberOfItems32	The number of failed logon attempts to the server. Can indicate whether password guessing programs are being used to crack the security on the server.
Errors Access Permissions	NumberOfItems32	The number of times opens on behalf of clients have failed with STATUS_ACCESS_DENIED. Can indicate whether somebody is randomly attempting to access files in hopes of getting at something that was not properly protected.
Errors Granted Access	NumberOfItems32	The number of times accesses to files opened successfully were denied. Can indicate attempts to access files without proper access authorization.

Category: Server Work Queues

CategoryType: MultiInstance.

The Server Work Queues performance object consists of counters that monitor the length of the queues and objects in the queues.

Instance: 0

Counter Name	Counter Type	Counter Description
Queue Length	NumberOfItems32	Queue Length is the current length of the server work queue for this CPU. A sustained queue length greater than four might indicate processor congestion. This is an instantaneous count, not an average over time.
Active Threads	NumberOfItems32	Active Threads is the number of threads currently working on a request from the server client for this CPU. The system keeps this number as low as possible to minimize unnecessary context switching. This is an instantaneous count for the CPU, not an average over time.
Available Threads	NumberOfItems32	Available Threads is the number of server threads on this CPU not currently working on requests from a client. The server dynamically adjusts the number of threads to maximize server performance.
Available Work Items	NumberOfItems32	Every request from a client is represented in the server as a 'work item,' and the server maintains a pool of available work items per CPU to speed processing. This is the instantaneous number of available work items for this CPU. A sustained near-zero value indicates the need to increase the MinFreeWorkItems registry value for the Server service. This value will always be 0 in the Blocking Queue instance.
Borrowed Work Items	NumberOfItems32	Every request from a client is represented in the server as a 'work item,' and the server maintains a pool of available work items per CPU to speed processing. When a CPU runs out of work items, it borrows a free work item from another CPU. An increasing value of this running counter might indicate the need to increase the 'MaxWorkItems' or 'MinFreeWorkItems' registry values for the Server service. This value will always be 0 in the Blocking Queue instance.
Work Item Shortages	NumberOfItems32	Every request from a client is represented in the server as a 'work item,' and the server maintains a pool of available work items per CPU to speed processing. A sustained value greater than zero indicates the need to increase the 'MaxWorkItems' registry value for the Server service. This value will always

Counter Name	Counter Type	Counter Description
		be 0 in the Blocking Queue instance.
Current Clients	NumberOfItems32	Current Clients is the instantaneous count of the clients being serviced by this CPU. The server actively balances the client load across all of the CPU's in the system. This value will always be 0 in the Blocking Queue instance.
Bytes Received/sec	RateOfCountsPerSecond64	The rate at which the Server is receiving bytes from the network clients on this CPU. This value is a measure of how busy the Server is.
Bytes Sent/sec	RateOfCountsPerSecond64	The rate at which the Server is sending bytes to the network clients on this CPU. This value is a measure of how busy the Server is.
Bytes Transferred/sec	RateOfCountsPerSecond64	The rate at which the Server is sending and receiving bytes with the network clients on this CPU. This value is a measure of how busy the Server is.
Read Operations/sec	RateOfCountsPerSecond64	Read Operations/sec is the rate the server is performing file read operations for the clients on this CPU. This value is a measure of how busy the Server is. This value will always be 0 in the Blocking Queue instance.
Read Bytes/sec	RateOfCountsPerSecond64	Read Bytes/sec is the rate the server is reading data from files for the clients on this CPU. This value is a measure of how busy the Server is.
Write Operations/sec	RateOfCountsPerSecond64	Write Operations/sec is the rate the server is performing file write operations for the clients on this CPU. This value is a measure of how busy the Server is. This value will always be 0 in the Blocking Queue instance.
Write Bytes/sec	RateOfCountsPerSecond64	Write Bytes/sec is the rate the server is writing data to files for the clients on this CPU. This value is a measure of how busy the Server is.
Total Bytes/sec	RateOfCountsPerSecond64	Total Bytes/sec is the rate the Server is reading and writing data to and from the files for the clients on this CPU. This value is a measure of how busy the Server is.
Total Operations/sec	RateOfCountsPerSecond64	Total Operations/sec is the rate the Server is performing file read and file write operations for the clients on this CPU. This value is a measure of how busy the Server is. This value will always be 0 in the Blocking Queue instance.
Context	RateOfCountsPerSecond3	Context Blocks Queued per second is the rate at which work

Counter Name	Counter Type	Counter Description
Blocks Queued/sec	2	context blocks had to be placed on the server's FSP queue to await server action.

Instance: 1

Instance: Blocking Queue

Category: ServiceModelEndpoint 3.0.0.0

CategoryType: MultiInstance.

ServiceModelEndpoint 3.0.0.0 performance counters

Category: ServiceModelEndpoint 4.0.0.0

CategoryType: MultiInstance.

ServiceModelEndpoint 4.0.0.0 performance counters

Category: ServiceModelOperation 3.0.0.0

CategoryType: MultiInstance.

ServiceModelOperation 3.0.0.0 performance counters

Category: ServiceModelOperation 4.0.0.0

CategoryType: MultiInstance.

ServiceModelOperation 4.0.0.0 performance counters

Category: ServiceModelService 3.0.0.0

CategoryType: MultiInstance.

ServiceModelService 3.0.0.0 performance counters

Category: ServiceModelService 4.0.0.0

CategoryType: MultiInstance.

Category: SMSvcHost 3.0.0.0

CategoryType: SingleInstance.

SMSvcHost 3.0.0.0 performance counters

Counter Name	Counter Type	Counter Description
Registrations Active for net.pipe	NumberOfItems3 2	The number of uri registrations currently active for net.pipe.
Registrations Active for net.tcp	NumberOfItems3 2	The number of uri registrations currently active for net.tcp.
Connections Accepted over net.pipe	NumberOfItems3 2	The total number of named pipe connections accepted over net.pipe.
Uris Registered for net.tcp	NumberOfItems3 2	The total number of uris that were succesfully registered for net.tcp.
Uris Unregistered for net.pipe	NumberOfItems3 2	The total number of uris that were succesfully unregistered for net.pipe.
Uris Unregistered for net.tcp	NumberOfItems3 2	The total number of uris that were succesfully unregistered for net.tcp.
Uris Registered for net.pipe	NumberOfItems3 2	The total number of uris that were succesfully registered for net.pipe.
Dispatch Failures over net.tcp	NumberOfItems3 2	The total number of failures dispatching messages received over net.tcp.
Protocol Failures over net.pipe	NumberOfItems3 2	The total number of failures at the protocol layer of net.pipe.
Protocol Failures over net.tcp	NumberOfItems3 2	The total number of failures at the protocol layer of net.tcp.
Dispatch Failures over net.pipe	NumberOfItems3 2	The total number of failures dispatching messages received over net.pipe.
Connections Accepted over net.tcp	NumberOfItems3 2	The total number of TCP connections accepted over net.tcp.

Counter Name	Counter Type	Counter Description
Connections Dispatched over net.pipe	NumberOfItems32	The total number of connections dispatched over net.pipe.
Connections Dispatched over net.tcp	NumberOfItems32	The total number of connections dispatched over net.tcp.

Category: SMSvcHost 4.0.0.0

CategoryType: SingleInstance.

SMSvcHost 4.0.0.0 performance counters

Counter Name	Counter Type	Counter Description
Registrations Active for net.pipe	NumberOfItems32	The number of uri registrations currently active for net.pipe.
Registrations Active for net.tcp	NumberOfItems32	The number of uri registrations currently active for net.tcp.
Connections Accepted over net.pipe	NumberOfItems32	The total number of named pipe connections accepted over net.pipe.
Uris Registered for net.tcp	NumberOfItems32	The total number of uris that were succesfully registered for net.tcp.
Uris Unregistered for net.pipe	NumberOfItems32	The total number of uris that were succesfully unregistered for net.pipe.
Uris Unregistered for net.tcp	NumberOfItems32	The total number of uris that were succesfully unregistered for net.tcp.
Uris Registered for net.pipe	NumberOfItems32	The total number of uris that were succesfully registered for net.pipe.
Dispatch Failures over net.tcp	NumberOfItems32	The total number of failures dispatching messages received over net.tcp.
Protocol Failures over net.pipe	NumberOfItems32	The total number of failures at the protocol layer of net.pipe.
Protocol Failures over net.tcp	NumberOfItems32	The total number of failures at the protocol layer of net.tcp.
Dispatch Failures over	NumberOfItems32	The total number of failures dispatching messages

Counter Name	Counter Type	Counter Description
net.pipe	2	received over net.pipe.
Connections Accepted over net.tcp	NumberOfItems32	The total number of TCP connections accepted over net.tcp.
Connections Dispatched over net.pipe	NumberOfItems32	The total number of connections dispatched over net.pipe.
Connections Dispatched over net.tcp	NumberOfItems32	The total number of connections dispatched over net.tcp.

Category: System

CategoryType: SingleInstance.

The System performance object consists of counters that apply to more than one instance of a component processors on the computer.

Counter Name	Counter Type	Counter Description
Processes	NumberOfItems32	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.
Threads	NumberOfItems32	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 execute instructions in a processor.
System Up Time	ElapsedTime	System Up Time is the elapsed time (in seconds) that the computer has been running since it was last started. This counter displays the difference between the start time and the current time.
Processor Queue Length	NumberOfItems32	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 need to divide this value by the number of processors servicing the workload. A sustained processor queue of less than 10 threads per processor is normally acceptable, dependent of the workload.
Floating	RateOfCountsPerSecond3	Floating Emulations/sec is the rate of floating emulations

Counter Name	Counter Type	Counter Description
Emulations/sec	2	performed by the system. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.
% Registry Quota In Use	RawFraction	% Registry Quota In Use is the percentage of the Total Registry Quota Allowed that is currently being used by the system. This counter displays the current percentage value only; it is not an average.
Alignment Fixups/sec	RateOfCountsPerSecond3 2	Alignment Fixups/sec is the rate, in incidents per seconds, at alignment faults were fixed by the system.
Exception Dispatches/sec	RateOfCountsPerSecond3 2	Exception Dispatches/sec is the rate, in incidents per second, at which exceptions were dispatched by the system.
File Data Operations/sec	RateOfCountsPerSecond3 2	File Data Operations/sec is the combined rate of read and write operations on all logical disks on the computer. This is the inverse of System: File Control Operations/sec. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.
File Control Operations/sec	RateOfCountsPerSecond3 2	File Control Operations/sec is the combined rate of file system operations that are neither reads nor writes, such as file system control requests and requests for information about device characteristics or status. This is the inverse of System: File Data Operations/sec and is measured in number of operations per second. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.
File Read Bytes/sec	RateOfCountsPerSecond6 4	File Read Bytes/sec is the overall rate at which bytes are read to satisfy file system read requests to all devices on the computer, including reads from the file system cache. It is measured in number of bytes per second. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.
File Read Operations/sec	RateOfCountsPerSecond3 2	File Read Operations/sec is the combined rate of file system read requests to all devices on the computer, including requests to read from the file system cache. It is measured in numbers of reads. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.
File Write Operations/sec	RateOfCountsPerSecond3 2	File Write Operations/sec is the combined rate of the file system write requests to all devices on the computer, including requests to write to data in the file system cache. It is

Counter Name	Counter Type	Counter Description
		measured in numbers of writes. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.
Context Switches/sec	RateOfCountsPerSecond32	Context Switches/sec is the combined rate at which all processors on the computer are switched from one thread to another. Context switches occur when a running thread voluntarily relinquishes the processor, is preempted by a higher priority ready thread, or switches between user-mode and privileged (kernel) mode to use an Executive or subsystem service. It is the sum of Thread\Context Switches/sec for all threads running on all processors in the computer and is measured in numbers of switches. There are context switch counters on the System and Thread objects. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.
System Calls/sec	RateOfCountsPerSecond32	System Calls/sec is the combined rate of calls to operating system service routines by all processes running on the computer. These routines perform all of the basic scheduling and synchronization of activities on the computer, and provide access to non-graphic devices, memory management, and name space management. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.
File Write Bytes/sec	RateOfCountsPerSecond64	File Write Bytes/sec is the overall rate at which bytes are written to satisfy file system write requests to all devices on the computer, including writes to the file system cache. It is measured in number of bytes per second. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.
File Control Bytes/sec	RateOfCountsPerSecond64	File Control Bytes/sec is the overall rate at which bytes are transferred for all file system operations that are neither reads nor writes, including file system control requests and requests for information about device characteristics or status. It is measured in numbers of bytes. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Category: TCP

CategoryType: SingleInstance.

The TCP performance object consists of counters that measure the rates at which TCP Segments are sent and received by using the TCP protocol. It includes counters that monitor the number of TCP connections in each TCP connection state.

Counter Name	Counter Type	Counter Description
Segments Received/sec	RateOfCountsPerSecond32	Segments Received/sec is the rate at which segments are received, including those received in error. This count includes segments received on currently established connections.
Connections Reset	NumberOfItems32	Connections Reset is the number of times TCP connections have made a direct transition to the CLOSED state from either the ESTABLISHED state or the CLOSE-WAIT state.
Segments Retransmitted/sec	RateOfCountsPerSecond32	Segments Retransmitted/sec is the rate at which segments are retransmitted, that is, segments transmitted containing one or more previously transmitted bytes.
Segments Sent/sec	RateOfCountsPerSecond32	Segments Sent/sec is the rate at which segments are sent, including those on current connections, but excluding those containing only retransmitted bytes.
Connection Failures	NumberOfItems32	Connection Failures is the number of times TCP connections have made a direct transition to the CLOSED state from the SYN-SENT state or the SYN-RCVD state, plus the number of times TCP connections have made a direct transition to the LISTEN state from the SYN-RCVD state.
Connections Established	NumberOfItems32	Connections Established is the number of TCP connections for which the current state is either ESTABLISHED or CLOSE-WAIT.
Segments/sec	RateOfCountsPerSecond32	Segments/sec is the rate at which TCP segments are sent or received using the TCP protocol.
Connections Passive	NumberOfItems32	Connections Passive is the number of times TCP connections have made a direct transition to the SYN-RCVD state from the LISTEN state.
Connections Active	NumberOfItems32	Connections Active is the number of times TCP connections have made a direct transition to the SYN-SENT state from the CLOSED state.

Category: Telephony

CategoryType: SingleInstance.

The Telephony System

Counter Name	Counter Type	Counter Description
Client Apps	NumberOfItems32	The number of applications that are currently using telephony services.
Incoming Calls/sec	RateOfCountsPerSecond32	The rate of incoming calls answered by this computer.
Current Incoming Calls	NumberOfItems32	Current incoming calls being serviced by this computer.
Current Outgoing Calls	NumberOfItems32	Current outgoing calls being serviced by this computer.
Outgoing Calls/sec	RateOfCountsPerSecond32	The rate of outgoing calls made by this computer.
Telephone Devices	NumberOfItems32	The number of telephone devices serviced by this computer.
Lines	NumberOfItems32	The number of telephone lines serviced by this computer.
Active Telephones	NumberOfItems32	The number of telephone devices that are currently being monitored.
Active Lines	NumberOfItems32	The number of telephone lines serviced by this computer that are currently active.

Category: Terminal Services

CategoryType: SingleInstance.

Terminal Services summary information.

Counter Name	Counter Type	Counter Description
Inactive Sessions	NumberOfItems32	Number of inactive Terminal Services sessions.
Active Sessions	NumberOfItems32	Number of active Terminal Services sessions.
Total Sessions	NumberOfItems32	Total number of Terminal Services sessions.

Category: Terminal Services Session

CategoryType: MultiInstance.

Terminal Services per-session resource monitoring.

Instance: RDP-Tcp 12

Counter Name	Counter Type	Counter Description
% Processor Time	Timer100Ns	% Processor Time is the percentage of elapsed time that all of process threads used the processor to execution instructions. An instruction is the basic unit of execution in a computer, a thread is the object that executes instructions, and a process is the object created when a program is run. Code executed to handle some hardware interrupts and trap conditions are included in this count.
% User Time	Timer100Ns	% User Time is the percentage of elapsed time that the process threads spent executing code in user mode. Applications, environment subsystems, and integral subsystems execute in user mode. Code executing in user mode cannot damage the integrity of the Windows executive, kernel, and device drivers. Unlike some early operating systems, Windows uses process boundaries for subsystem protection in addition to the traditional protection of user and privileged modes. Some work done by Windows on behalf of the application might appear in other subsystem processes in addition to the privileged time in the process.
% Privileged Time	Timer100Ns	% Privileged Time is the percentage of elapsed time that the process threads spent executing code in privileged mode. When a Windows system service is called, the service will often run in privileged mode to gain access to system-private data. Such data is protected from access by threads executing in user mode. Calls to the system can be explicit or implicit, such as page faults or interrupts. Unlike some early operating systems, Windows uses process boundaries for subsystem protection in addition to the traditional protection of user and privileged modes. Some work done by Windows on behalf of the application might appear in other subsystem processes in addition to the privileged time in the process.
Virtual Bytes Peak	NumberOfItems64	Virtual Bytes Peak is the maximum size, in bytes, of virtual address space the process has used at any one time. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. However, virtual space is finite, and the process might limit its ability to load libraries.

Counter Name	Counter Type	Counter Description
Virtual Bytes	NumberOfItems64	Virtual Bytes is the current size, in bytes, of the virtual address space the process is using. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. Virtual space is finite, and the process can limit its ability to load libraries.
Page Faults/sec	RateOfCountsPerSecond32	Page Faults/sec is the rate at which page faults by the threads executing in this process are occurring. A page fault occurs when a thread refers to a virtual memory page that is not in its working set in main memory. This may not cause the page to be fetched from disk if it is on the standby list and hence already in main memory, or if it is in use by another process with whom the page is shared.
Working Set Peak	NumberOfItems64	Working Set Peak is the maximum size, in bytes, of the Working Set of this process at any point in time. The Working Set is the set of memory pages touched recently by the threads in the process. If free memory in the computer is above a threshold, pages are left in the Working Set of a process even if they are not in use. When free memory falls below a threshold, pages are trimmed from Working Sets. If they are needed they will then be soft-faulted back into the Working Set before they leave main memory.
Working Set	NumberOfItems64	Working Set is the current size, in bytes, of the Working Set of this process. The Working Set is the set of memory pages touched recently by the threads in the process. If free memory in the computer is above a threshold, pages are left in the Working Set of a process even if they are not in use. When free memory falls below a threshold, pages are trimmed from Working Sets. If they are needed they will then be soft-faulted back into the Working Set before leaving main memory.
Page File Bytes Peak	NumberOfItems64	Page File Bytes Peak is the maximum number of bytes this process has used in the paging file(s). Paging files are used to store pages of memory used by the process that are not contained in other files. Paging files are shared by all processes, and the lack of space in paging files can prevent other processes from allocating memory.
Page File Bytes	NumberOfItems64	Page File Bytes is the current number of bytes that this process has used in the paging file(s). Paging files are used to store pages of memory used by the process that are not contained in other files. Paging files are shared by all processes, and the lack of space in paging files can prevent other processes from allocating memory.
Private Bytes	NumberOfItems64	Private Bytes is the current size, in bytes, of memory that this process has allocated that cannot be shared with other

Counter Name	Counter Type	Counter Description
		processes.
Thread Count	NumberOfItems32	The number of threads currently active in this process. An instruction is the basic unit of execution in a processor, and a thread is the object that executes instructions. Every running process has at least one thread.
Pool Paged Bytes	NumberOfItems32	Pool Paged Bytes is the size, in bytes, of the paged pool, an area of system memory (physical memory used by the operating system) for objects that can be written to disk when they are not being used. Memory\Pool Paged Bytes is calculated differently than Process\Pool Paged Bytes, so it might not equal Process\Pool Paged Bytes_Total. This counter displays the last observed value only; it is not an average.
Pool Nonpaged Bytes	NumberOfItems32	Pool Nonpaged Bytes is the size, in bytes, of the nonpaged pool, an area of system memory (physical memory used by the operating system) for objects that cannot be written to disk, but must remain in physical memory as long as they are allocated. Memory\Pool Nonpaged Bytes is calculated differently than Process\Pool Nonpaged Bytes, so it might not equal Process\Pool Nonpaged Bytes_Total. This counter displays the last observed value only; it is not an average.
Handle Count	NumberOfItems32	The total number 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.
Input WdBytes	RateOfCountsPerSecond32	Number of bytes input on this session after all protocol overhead has been removed.
Input WdFrames	RateOfCountsPerSecond32	The number of frames input after any additional protocol added frames have been removed.
Input WaitForOutBuf	NumberOfItems32	The number of times that a wait for an available send buffer was done by the protocols on the client side of the connection.
Input Frames	RateOfCountsPerSecond32	Number of frames (packets) input on this Session.
Input Bytes	RateOfCountsPerSecond32	Number of bytes input on this session that includes all protocol overhead.
Input Compressed Bytes	RateOfCountsPerSecond32	Number of bytes input after compression. This number compared with the Total Bytes input is the compression

Counter Name	Counter Type	Counter Description
		ratio.
Input Compress Flashes	NumberOfItems32	Number of input compression dictionary flushes. When the data can not be compressed, the compression dictionary is flushed so that newer data has a better chance of being compressed. Some causes of data not compressing includes transferring compressed files over Client Drive Mapping.
Input Errors	NumberOfItems32	Number of input errors of all types. Some example input errors are lost ACK's, badly formed packets, etc.
Input Timeouts	NumberOfItems32	The total number of timeouts on the communication line as seen from the client side of the connection. These are typically the result of a noisy line. On some high latency networks, this could be the result of the protocol timeout being too short. Increasing the protocol timeout on these types of lines will improve performance by reducing needless re-transmissions.
Input Async Frame Error	NumberOfItems32	Number of input async framing errors. These can be caused by a noisy transmission line. Using a smaller packet size may help in some cases.
Input Async Overrun	NumberOfItems32	Number of input async overrun errors. These can be caused by the baud rate being faster than the computer can handle, or a non-16550 serial line is used. Overruns can also occur if too many high speed serial lines are active at one time for the processor's power.
Input Async Overflow	NumberOfItems32	Number of input async overflow errors. These can be caused by a lack of buffer space available on the host.
Input Async Parity Error	NumberOfItems32	Number of input async parity errors. These can be caused by a noisy transmission line
Input Transport Errors	NumberOfItems32	Number of Terminal Services transport-level errors on input.
Output WdBytes	RateOfCountsPerSecond32	Number of bytes output on this session after all protocol overhead has been removed.
Output WdFrames	RateOfCountsPerSecond32	The number of frames output before any additional protocol frames have been added.
Output WaitForOutBuf	NumberOfItems32	This is the number of times that a wait for an available send buffer was done by the protocol on the server side of

Counter Name	Counter Type	Counter Description
		the connection.
Output Frames	RateOfCountsPerSecond32	Number of frames (packets) output on this session.
Output Bytes	RateOfCountsPerSecond32	Number of bytes output on this Session that includes all protocol overhead.
Output Compressed Bytes	RateOfCountsPerSecond32	Number of bytes output after compression. This number compared with the Total Bytes output is the compression ratio.
Output Compress Flushes	NumberOfItems32	Number of output compression dictionary flushes. When the data can not be compressed, the compression dictionary is flushed so that newer data has a better chance of being compressed. Some causes of data not compressing includes transferring compressed files over Client Drive Mapping.
Output Errors	NumberOfItems32	Number of output errors of all types. Some example output errors are lost ACK's, badly formed packets, etc.
Output Timeouts	NumberOfItems32	The total number of timeouts on the communication line from the host side of the connection. These are typically the result of a noisy line. On some high latency networks, this could be the result of the protocol timeout being too short. Increasing the protocol timeout on these types of lines will improve performance by reducing needless re-transmissions.
Output Async Frame Error	NumberOfItems32	Number of output async framing errors. This could be caused by a hardware or line problem.
Output Async Overrun	NumberOfItems32	Number of output async overrun errors.
Output Async Overflow	NumberOfItems32	Number of output async overflow errors.
Output Async Parity Error	NumberOfItems32	Number of output async parity errors. These can be caused by a hardware or line problem.
Output Transport Errors	NumberOfItems32	Number of Terminal Services transport-level errors on output.
Total WdBytes	RateOfCountsPerSecond32	Total number of bytes on this Session after all protocol overhead has been removed.

Counter Name	Counter Type	Counter Description
Total WdFrames	RateOfCountsPerSecond32	The total number of frames input and output before any additional protocol frames have been added.
Total WaitForOutBuf	NumberOfItems32	The number of times that a wait for an available send buffer was done by the protocols on both the server and client sides of the connection.
Total Frames	RateOfCountsPerSecond32	Total number of frames (packets) on this Session.
Total Bytes	RateOfCountsPerSecond32	Total number of bytes on this Session that includes all protocol overhead.
Total Compressed Bytes	RateOfCountsPerSecond32	Total number of bytes after compression. This number compared with the total bytes is the compression ratio.
Total Compress Flushes	NumberOfItems32	Total number of compression dictionary flushes. When the data can not be compressed, the compression dictionary is flushed so that newer data has a better chance of being compressed. Some causes of data not compressing includes transferring compressed files over Client Drive Mapping.
Total Errors	NumberOfItems32	Total number of errors of all types. Some example errors are lost ACK's, badly formed packets, etc.
Total Timeouts	NumberOfItems32	The total number of timeouts on the communication line from both the host and client sides of the connection. These are typically the result of a noisy line. On some high latency networks, this could be the result of the protocol timeout being too short. Increasing the protocol timeout on these types of lines will improve performance by reducing needless re-transmissions.
Total Async Frame Error	NumberOfItems32	Total number of async framing errors. These can be caused by a noisy transmission line. Using a smaller packet size may help in some cases.
Total Async Overrun	NumberOfItems32	Total number of async overrun errors. These can be caused by the baud rate being faster than the computer can handle, or a non-16550 serial line is used. Overruns can also occur if too many high speed serial lines are active at one time for the processor's power.
Total Async Overflow	NumberOfItems32	Total number of async overflow errors. These can be caused by a lack of buffer space available on the host.

Counter Name	Counter Type	Counter Description
Total Async Parity Error	NumberOfItems32	Total number of async parity errors. These can be caused by a noisy transmission line.
Total Transport Errors	NumberOfItems32	Total number of Terminal Services transport-level errors.
Total Protocol Cache Reads	NumberOfItems32	Total references to all protocol caches.
Total Protocol Cache Hits	NumberOfItems32	Total hits in all protocol caches. The protocol caches Windows objects that are likely to be re-used to avoid having to re-send them on the transmission line. Example objects are Windows icons and brushes. Hits in the cache represent objects that did not need to be re-sent.
Total Protocol Cache Hit Ratio	NumberOfItems32	Overall hit ratio for all protocol caches.
Protocol Bitmap Cache Reads	NumberOfItems32	Number of references to the protocol bitmap cache.
Protocol Bitmap Cache Hits	NumberOfItems32	Number of hits in the protocol bitmap cache.
Protocol Bitmap Cache Hit Ratio	NumberOfItems32	Hit ratio in the protocol bitmap cache. A higher hit ratio means better performance since data transmissions are reduced. Low hit ratios are due to the screen updating with new information that is either not re-used, or is flushed out of the client cache.
Protocol Glyph Cache Reads	NumberOfItems32	Number of references to the protocol glyph cache.
Protocol Glyph Cache Hits	NumberOfItems32	Number of hits in the protocol glyph cache.
Protocol Glyph Cache Hit Ratio	NumberOfItems32	Hit ratio in the protocol glyph cache. A higher hit ratio means better performance since data transmissions are reduced. Low hit ratios are due to the screen updating with new information that is either not re-used, or is flushed out of the client cache.
Protocol Brush Cache Reads	NumberOfItems32	Number of references to the protocol brush cache.
Protocol Brush Cache Hits	NumberOfItems32	Number of hits in the protocol brush cache.

Counter Name	Counter Type	Counter Description
Protocol Brush Cache Hit Ratio	NumberOfItems32	Hit ratio in the protocol brush cache. A higher hit ratio means better performance since data transmissions are reduced. Low hit ratios are due to the screen updating with new information that is either not re-used, or is flushed out of the client cache.
Protocol Save Screen Bitmap Cache Reads	NumberOfItems32	Number of references to the protocol save screen bitmap cache.
Protocol Save Screen Bitmap Cache Hits	NumberOfItems32	Number of hits in the protocol save screen bitmap cache.
Protocol Save Screen Bitmap Cache Hit Ratio	NumberOfItems32	Hit ratio in the protocol save screen bitmap cache. A higher hit ratio means better performance since data transmissions are reduced. Low hit ratios are due to the screen updating with new information that is either not re-used, or is flushed out of the client cache.
Input Compression Ratio	NumberOfItems32	Compression ratio of the server input data stream.
Output Compression Ratio	NumberOfItems32	Compression ratio of the server output data stream.
Total Compression Ratio	NumberOfItems32	Total compression ratio of the server data stream.

Instance: Console

Instance: RDP-Tcp 11

Category: Thread

CategoryType: MultiInstance.

The Thread performance object consists of counters that measure aspects of thread behavior. A thread is the basic object that executes instructions on a processor. All running processes have at least one thread.

Instance: iexplore/1#2

Counter Name	Counter Type	Counter Description
% Processor Time	Timer100Ns	% Processor Time is the percentage of elapsed time that all of process threads used the processor to execution instructions. An instruction is the basic unit of execution in a computer, a thread is the object that executes instructions, and a process is the object created when a program is run. Code executed to handle some hardware interrupts and trap conditions are included in this count.
% User Time	Timer100Ns	% User Time is the percentage of elapsed time that this thread has spent executing code in user mode. Applications, environment subsystems, and integral subsystems execute in user mode. Code executing in user mode cannot damage the integrity of the Windows NT Executive, Kernel, and device drivers. Unlike some early operating systems, Windows NT uses process boundaries for subsystem protection in addition to the traditional protection of user and privileged modes. These subsystem processes provide additional protection. Therefore, some work done by Windows NT on behalf of your application might appear in other subsystem processes in addition to the privileged time in your process.
% Privileged Time	Timer100Ns	% Privileged Time is the percentage of elapsed time that the process threads spent executing code in privileged mode. When a Windows system service is called, the service will often run in privileged mode to gain access to system-private data. Such data is protected from access by threads executing in user mode. Calls to the system can be explicit or implicit, such as page faults or interrupts. Unlike some early operating systems, Windows uses process boundaries for subsystem protection in addition to the traditional protection of user and privileged modes. Some work done by Windows on behalf of the application might appear in other subsystem processes in addition to the privileged time in the process.
Context Switches/sec	RateOfCountsPerSecond32	Context Switches/sec is the rate of switches from one thread to another. Thread switches can occur either inside of a single process or across processes. A thread switch can be caused either by one thread asking another for information, or by a thread being preempted by another, higher priority thread becoming ready to run. Unlike some early operating systems, Windows NT uses process boundaries for subsystem protection in addition to the traditional protection of user and privileged modes. These subsystem processes provide additional protection. Therefore, some work done by Windows NT on behalf of an application appear in other subsystem processes in addition to the privileged time in the application. Switching to the subsystem process causes one Context Switch in the application thread. Switching back

Counter Name	Counter Type	Counter Description
		causes another Context Switch in the subsystem thread.
Elapsed Time	ElapsedTime	The total elapsed time (in seconds) this thread has been running.
Priority Current	NumberOfItems32	The current dynamic priority of this thread. The system can raise the thread's dynamic priority above the base priority if the thread is handling user input, or lower it towards the base priority if the thread becomes compute bound.
Priority Base	NumberOfItems32	The current base priority of this thread. The system can raise the thread's dynamic priority above the base priority if the thread is handling user input, or lower it towards the base priority if the thread becomes compute bound.
Start Address	NumberOfItemsHEX32	Starting virtual address for this thread.
Thread State	NumberOfItems32	Thread State is the current state of the thread. It is 0 for Initialized, 1 for Ready, 2 for Running, 3 for Standby, 4 for Terminated, 5 for Wait, 6 for Transition, 7 for Unknown. A Running thread is using a processor; a Standby thread is about to use one. A Ready thread wants to use a processor, but is waiting for a processor because none are free. A thread in Transition is waiting for a resource in order to execute, such as waiting for its execution stack to be paged in from disk. A Waiting thread has no use for the processor because it is waiting for a peripheral operation to complete or a resource to become free.
Thread Wait Reason	NumberOfItems32	Thread Wait Reason is only applicable when the thread is in the Wait state (see Thread State). It is 0 or 7 when the thread is waiting for the Executive, 1 or 8 for a Free Page, 2 or 9 for a Page In, 3 or 10 for a Pool Allocation, 4 or 11 for an Execution Delay, 5 or 12 for a Suspended condition, 6 or 13 for a User Request, 14 for an Event Pair High, 15 for an Event Pair Low, 16 for an LPC Receive, 17 for an LPC Reply, 18 for Virtual Memory, 19 for a Page Out; 20 and higher are not assigned at the time of this writing. Event Pairs are used to communicate with protected subsystems (see Context Switches).
ID Process	NumberOfItems32	ID Process is the unique identifier of this process. ID Process numbers are reused, so they only identify a process for the lifetime of that process.
ID Thread	NumberOfItems32	ID Thread is the unique identifier of this thread. ID Thread numbers are reused, so they only identify a thread for the

Counter Name	Counter Type	Counter Description
		lifetime of that thread.

Instance: iexplore/7#1

Instance: winlogon/5#1

... and many more.

Category: UDP

CategoryType: SingleInstance.

The UDP performance object consists of counters that measure the rates at which UDP datagrams are sent and received by using the UDP protocol. It includes counters that monitor UDP protocol errors.

Counter Name	Counter Type	Counter Description
Datagrams Received Errors	NumberOfItems32	Datagrams Received Errors is the number of received UDP datagrams that could not be delivered for reasons other than the lack of an application at the destination port.
Datagrams Sent/sec	RateOfCountsPerSecond32	Datagrams Sent/sec is the rate at which UDP datagrams are sent from the entity.
Datagrams No Port/sec	RateOfCountsPerSecond32	Datagrams No Port/sec is the rate of received UDP datagrams for which there was no application at the destination port.
Datagrams/sec	RateOfCountsPerSecond32	Datagrams/sec is the rate at which UDP datagrams are sent or received by the entity.
Datagrams Received/sec	RateOfCountsPerSecond32	Datagrams Received/sec is the rate at which UDP datagrams are delivered to UDP users.

Category: WF (System.Workflow) 4.0.0.0

CategoryType: MultiInstance.

Windows Workflow Foundation Performance Counters

Category: Windows Workflow Foundation

CategoryType: MultiInstance.

Category: WMI Objects

CategoryType: SingleInstance.

Number of WMI High Performance provider returned by WMI Adapter

Counter Name	Counter Type	Counter Description
HiPerf Validity	NumberOfItems3 2	Shows if High Performance Classes are valid
HiPerf Classes	NumberOfItems6 4	Shows High Performance Classes