



# Using Perfmon with Domino for IBM Netfinity Servers

*By Gary Sullivan*

*MT Consultant*

*IBM Advanced Technical Support*

## **Abstract**

---

*This White Paper focuses on use of the NT Performance Monitor tool (Perfmon) with Lotus Domino Netfinity servers. This paper is organized in three major areas. The first area provides an overview of Perfmon constructs, followed by sections covering the chart, log and report functions as well as detail on Perfmon overhead. The second area addresses using Perfmon data to explore potential server bottlenecks, taking a resource oriented approach. The third area focuses on using Perfmon with Domino statistics to conduct correlation analysis.*

The Performance Monitor tool, commonly called perfmon, is available with the NT operating system and can be a valuable tool for monitoring your Domino Netfinity Server. Perfmon is a graphical utility and offers many capabilities. Perfmon can

- ◆ View data from multiple systems at the same time
- ◆ Log data from different computers
- ◆ Change current activity charts on the fly
- ◆ Show the results of changes
- ◆ Trigger an action when a threshold is exceeded
- ◆ Combine log files for long term storage
- ◆ Save “quick start” options

*To Invoke Perfmon, click on start, then select programs, then administrative tools, then Performance Monitor. Alternatively, You can type PERFMON on the command Line or set up an ICON to invoke the program.*

As you can see, Perfmon offers a rich set of features, many of which will be covered here. We will specifically explore the chart, log and report functions.

Chart provides a graphical mode to monitor the current activity on your Netfinity server. Log provides the capability to record activity on your server(s) for review at a later time using either the chart facility or the report facility. The report function provides the capability to view specific numerical values for individual counters such as the CPU utilization percentage.

## Performance Monitor Objects

Perfmon provides a broad set of both hardware and software objects to monitor. All of this information is organized in a hierarchy. Systems contain objects within which are instances of the object and within the object instance are counters.

For example, once we select the TBsut1 server to monitor, we can select the object(s) to monitor. In this example, we select the Processor object to gather metrics on the systems CPU(s). Next is the instance list. The instance list will show all instances of the selected object. If our server has only one CPU, then the instance list will have only one entry, “0”. If we have a dual processor server, then the instance list would have “0” and “1” from which to select either CPU 0 or CPU 1 or both.

Once instance choices (if any) are complete, the counter of interest can be selected. In this example, “% Processor Time” is a good choice to monitor the percent of the time the CPU is busy.

Note that in the case where the computer being monitored has more than one

*Once you select an object, the most important counter is automatically highlighted. So, it is generally wise to select the highlighted counter.*

component of the same object type, there may also be a “\_Total” instance to represent all the values of all instances of the object. Normally this value will be a sum.

Some objects will only appear if the process or service represented by the object is installed. A base set of objects, sometimes referred to as core objects, always show up in Perfmon. The base or core set of objects include:

Object	Object Description
System	Set of counters for all system hardware and software
Memory	Random access memory used to store programs and data
Cache	Memory that holds recently accessed data for quicker access
Paging File	File used to hold excess virtual memory allocations
Processor	Hardware unit (CPU) that executes instructions
Process	Software unit that represents an executing program
Thread	Unit of execution - part of a process that uses the processor
PhysicalDisk	Hardware disk - spindle or RAID device
LogicalDisk	Logical views of disk space, disk partitions
Redirector	File system to handle file requests to network servers
Objects	System software objects

Pe  
rf  
or  
Objects

## Performance Monitor Counters

Each object in Perfmon has a set of counters associated with it. A counter provides definition for the type of data that is available for the type of object selected. When a Perfmon counter is added to a chart, report or a log, it becomes activated and data is collected. In fact, when any counter is selected, all counters for the selected object are activated and data is collected for all counters for that object but only the selected counters are displayed.

Perfmon has three types of counters:

- **Averaging Counters** show the average value. Note that when counters of this type are started, there is an interval before any data values are displayed. This wait is due to the need to have multiple values in order to take an average.
- **Instantaneous Counters** simply show the most recent data value.
- **Difference Counters** subtract the current measurement from the previous one and show the difference value. If this value is negative, Perfmon will only display a zero.

*To display the definition for any counter, click on the **Explain** button in the **Add To Chart** box. The explain function will provide the Perfmon definition for the selected counter. Note that this is a good way to become familiar with each of the counters.*

## Chart Function

One of the most visible and possibly the most useful view in Perfmon is provided by the chart function.

*To access the chart function, select the view menu pull down and click on **CHART**.*

The chart view's graphic mode can be used to monitor the current (real time) activity on your Netfinity server. A chart will display the value of selected counters across time. One or more counters can be displayed at a time.

To add a counter to a chart from the chart view, click on edit, then click on **Add To Chart** (or click on the + at the top of the screen) and in the

**Add To Chart** window, select the server you want to monitor and then select the objects and counters you want to have displayed. You can monitor not only local Netfinity servers but also

*If you want to monitor a remote NT server, you must have a user ID and a password in the administrator group for that system.*

remote Netfinity servers. To monitor a remote server, type the UNC name (for example \\TBSut1) of the computer that you want to monitor in the computer field of the **Add To Chart** window. The **Select Computer** button to the right of this field also lets you select a remote Netfinity server.

In the options menu, click on **Chart** to view the chart options menu. Here you may change many of the chart view properties. In addition to the regular values one might expect to be present in these options, is the Update time interval. The default interval is 1 second. Perfmon dynamically updates the chart on this interval. Larger intervals provide for more time span in the chart and less overhead due to Perfmon screen update activity. This choice is a trade off between the granularity of the data and the overhead of Perfmon. Many analysts find that a 5 to 15 second interval is adequate.

As data values are added to the chart, the thick vertical line that represents "now" on the timeline, moves to the right. The most recent data value is displayed just to the left of this "now" line and the oldest data value is just to the right.

*With many counters added to a chart, it becomes difficult to follow a single counter. Selecting a counter and then pressing the **CTRL+H** keys will cause Perfmon to highlight the selected counter, and it will be much easier to discern on the chart. Backspace will also highlight a counter.*

Perfmon provides a handy facility for saving chart settings. From the file menu pulldown, click on **Save Chart Settings As**, specify the folder and filename you desire and then click **Save**. Note that this action saves the chart settings (not the data) for use in a subsequent session. This facility is quite useful for creating sets of object counters to monitor.

*Performance Monitor chart settings files have a **.PMC** file extension.*

## Log Function

Log is a useful function of Perfmon providing the capability to capture and record activity from your systems to disk for review and analysis at a later time. Once captured, log data can be explored using the chart function, the report function or can be converted to an export format for importation into a spread sheet. Export options include a tab-delimited (.tsv) or comma delimited (.csv) text file. To export data, select File, then the Export option.

*It is often instructive to collect logs periodically and keep them for use in comparisons when analyzing data from a period in which a problem exists.*

The process for selecting objects to be added to the log file is quite similar to that for adding counters to the chart function. The major difference is that the logging function does not allow for selection of individual counters. All counters within the selected object are logged.

*To access the log function, select the view menu pull down and click on **LOG**.*

To add objects to the log file, first click **New Log Settings** in the file menu. This action will clear out any existing log settings. Then on the edit menu pulldown, click on **Add to Log**. This action will display the **Add to Log** window which has the now familiar selection for the server to monitor and for the objects to log. The capability to select multiple objects and ranges of objects exists using the standard CTRL and SHIFT keys.

When finished selecting computers and objects, click on **Done**. Then choose **Log** from the Options menu. In the options box, you indicate the drive, folder and filename for the log file ( a .LOG file type), and the sampling interval (the default is 15 seconds). Again, the selection of the interval is a trade off between the granularity of the data and the size of the log file. In most cases 15 seconds is fine.

*To add objects to the log from other systems, simply select the new system in the computer field and select the objects to be logged from that system.*

Once these selections are complete, click on the **Start Log** button to start the logging process. The Perfmon window will show the status of the log file that is being created, including each server selected and the objects that are being collected from that system.

*To save the log settings, click on **Save Log Setting** As on the **File** menu pulldown. Perfmon log setting files have .PML file extension and save only the settings not the data.*

While logging is active, the bookmark option becomes available. Bookmarks allow inserting of comments and are time tagged to allow easy access to points in time in the log.

Once enough data has been logged, stop the log by selecting the **Options** menu pulldown, then click on **log** and then select **Stop Log**. Logging to the same file a second time will result in appending data to the log file.

To view the data in the log file, you must stop logging to it and the file must be closed. Opening a log that is collecting will stop the log and clear all the counter settings. Perfmon does not allow for using the Chart or Report view to see a running log because the views share the same data source.

*To see a running log, start a second copy of Perfmon and select **Data From** to be the running log.*

Once the log file is available, you can access it from the chart or report view. By default, Perfmon displays “current” data. To view the log data, you must switch to the log as the source of data. On the Option menu pulldown, click on **Data From**. In the **Data From** window, select the Log File option and type in or search for the log file name.

Once the data source is set, on the Edit menu pulldown, click on the **Time Window**. This action will display the **Input Log File Timeframe** window. In this box, the times for the log data are displayed graphically and feature slide buttons at the start and end of the data timeline. You can easily slide the button bars to select the time range that you are interested in viewing or possibly relogging. When you are satisfied with the time selection, click on **OK**.

*If the log file contains Bookmarks, you can select these from the bookmarks section of the window and click to set the mark as the start or stop point of the time range.*

To display the collected log data which is now the “current” data, follow the normal steps to add systems and counters to a chart or report view. Note that the selections available to add are defined by the data contained in the log.

Logging data allows for collecting/viewing data over a long period. This capability allows the analyst to detect trends and enables a measure of capacity planning based on those trends.

Collection of Perfmon data can be automated using the NT **AT** command to start and stop the monitor. Time intervals should be selected to capture data from periods of peak activity. Here is an example of the commands need to automate data collection from 9am till 11am, Monday thru Friday:

```
at \\TBsut1 09:00 /date:M,T,W,Th,F “monitor START”
```

```
at \\TBsut1 11:00 /date:M,T,W,Th,F “monitor STOP”
```

*Note that the schedule service must be running for the **AT** command to function. It is also necessary to create a settings file to specify the objects and interval for logging by using the save workspace settings option.*

If implemented, central monitoring will permit the data from multiple systems to be stored in a single log. Be aware that using centralized monitoring does increase network traffic. Collecting data on each server and transferring the data at an off peak time may be a better option but will result in multiple log files to manage.

Log files can be quite large. Collecting data from peak periods only and judiciously selecting the logging interval can minimize the disk resources required. Another viable technique involves using

the relogging function to reduce the amount of data saved. Also of utility is the capability to log data multiple times. The original logged data can be “logged” a second time.

While relogging the data, increase the log option and update time interval. If the log file was originally produced with a 30 second interval, relogging the data with a 60 second interval will reduce the disk requirement by 50 percent.

Since relogging with a larger interval will reduce the level of granularity, this action is normally done after the data has been

*To increase the Log Option update time interval, select **DATA FROM** on the **Options** menu.*

reviewed at the original collection interval. This technique can be used to append data to an archive file that summarizes the historical data.

## Report Function

The report view is useful for displaying the numeric data behind the individual Perfmon counters.

Perfmon provides the ability to generate reports that show the detailed data from either the log data or from “current” data. In most cases, report will be used to help analyze data from a log file

*To access the Report function, select the view menu pull down and click on **Report**.*

but it can also be used to create a point in time snapshot of current performance data.

To add a counter to a report from the report view, click on **Add To Report** and in the **Add To Report** window, select the system, object and counters desired,

then click on **Add**. When you have selected all the counters desired, click on **Done**. The report will show the System/object/instance/counter hierarchy for all the selected

*Perfmon report settings files have a .PMR file type.*

counters. To save your report settings, click on **Save Report Settings As** from the **File** menu pulldown. Note that only the report settings are saved, not the report data.

If the report is using “current” data (as opposed to using a log file), the report update frequency can be set on the options menu.

Alternatively, specify manual update and when an update to the report contents is desired, click on

*Looking to print? Unfortunately, Perfmon does not provide a print function. To print charts or reports, copy the screen to the Clipboard and use another application to print the clipboard contents.*

**Options** and select **Update Now**.

## Perfmon Overhead

The overhead of using Perfmon is not normally a large concern since there is little other choice if one is to have access to the needed data. However, there are some things to be aware of to run perfmon in the most efficient manor.

Perfmon cannot monitor disk activity unless the **DISKPERF -Y** or **-YE** command is

*A large portion of the overhead caused by Perfmon is writing to the screen and/or log files.*

issued and the server is rebooted. The **-YE** option enables measurement of physical disks in a RAID stripe set. By setting this monitoring feature on, additional load is placed on the server. If this load is a concern, you can enable the feature to gather data and then disable it with the **DISKPERF -N** command.

*To enable monitoring disk activity on a remote server use **DISKPERF -Y \\TBsut1** then restart the remote system*

While Perfmon was designed to have minimal impact, setting a high (very frequent) sampling rate can have a noticeable impact. Pick the least frequent interval that will provide the level of granularity in the data that is needed.

Log file size has been previously mentioned in the Log Function section but it is worth mentioning again. Just remember to select the monitoring period and logging interval judiciously to minimize the disk resources required. Use the relogging function to reduce the amount of data saved long term.

If Perfmon is running on the server that is to be analyzed then the resources on that server will be affected by the local execution of Perfmon. Centralizing data collection by running Perfmon remotely minimizes these impacts at the cost of increased network traffic. This centralized approach also allows the collected data to be placed in a single log file thus reducing the number of files to be dealt with and simplifying data management.

*Since log files can be large and active, they can also skew the measurement data.*

## Using Perfmon Data

There are two general categories of activities that will involve Perfmon metrics. These categories are capacity planning and performance analysis. This paper will deal with using Perfmon data to do server level performance analysis.

There are many reasons to engage in performance analysis. In many cases concerns about system responsiveness initiate this effort which centers around determining how resources are handling the work that is requested of them, in order to understand the bottlenecks the server is encountering.

An important concept to understand is that every system has a bottleneck. The “trick” is to choose the bottleneck with which you are most comfortable. In most cases this “desired” bottleneck is going to be the processor(s). There are two rational for this assertion:

- Processor utilization is generally well understood by management and thus provides the most direct justification for a server upgrade.
- Processor utilization is relatively easy to quantify.



In general, bottlenecks can be grouped in terms of the associated resources and so can be considered in one of four categories:

1. Processor
2. Memory
3. Disk
4. Network

*The resource that is restricting the flow of work thru the server or slowing system responsiveness is termed “the bottleneck”. It is not uncommon for one bottleneck to be resolved only to have another one surface.*

The exceptions to this

generalization arise in situations where one resource area can negatively impact another. For example, inadequate real memory can generate extra load on the processor as well as create additional disk activity. Many situations arise where it is wise to focus on examination of several metrics in concert to provide the most complete picture. The reason for this assertion (although we approach these topics as individual areas), is that all of a servers resources are interrelated. In most situations you should focus on a set of key related metrics.

## Processor

To assess whether the processor is a bottleneck or not, there are several Perfmon counters that are of interest.

**Processor:%Processor time** shows the percent of time that the processor is not running the idle thread.

**Processor:%User Time** shows the percent of time that the processor is performing user services.

**Processor:%Privileged Time** shows the percent of time that the processor is executing NT system services.

**Processor: Interrupts per second** shows the number of interrupts the processor is handling from applications or from hardware devices.

**System: Processor Queue Length** shows the depth of requests in the processor queue. All processors use this single queue in which threads wait for processor cycles.

**Server Work Queues: Queue Length** shows the depth of the requests in the queue for the selected processor.

Many performance analysis practitioners will assert that when the processor is consistently running over a given percent busy, the processor has become a bottleneck. In reality, this assertion may not be true although a high utilization level is certainly cause to monitor the processor closely because it indicates that queuing is likely occurring.

In addition, the server should be checked to make sure that excessive paging is not occurring or that interrupt levels are not excessive.

*Using complex screen savers or inadequately sizing the pagefile.sys paging file can also contribute to unnecessary processor utilization.*

If there are no problems with real memory shortage or excessive interrupts, then a determination of whether the system is suffering excessive queuing is in order. This can be accomplished by examining queue lengths. If the processor is 90 % busy but the queue lengths are 1 or 2, then the processor (though on the verge of excessive queuing) can be expected to service requests without abnormal delay.

*High Processor utilization combined with long queue lengths is a strong indication of a processor bottleneck.*

If the processor is a bottleneck, then an upgrade or replacement is in order. If new or additional hardware is not possible, consider offloading some work to another Netfinity server or possibly adjusting the priorities of the processes on the system to favor important work to minimize the impact of the bottleneck. To change a process's priority, go to the Task Manager Processes option, right click on the process whose priority you wish to adjust, choose SetPriority and then select from the Realtime, High, Normal or Low options. Note that the Realtime option can result in a process hogging the CPU and so should be used with care.

## Memory

Possibly the most common bottleneck on Netfinity servers is real memory. To access the processor (which is a prereq to any action), a process must first occupy memory thus adequate memory is essential for a productive server. To assess whether real memory is a bottleneck or not, there are Perfmon counters of interest.

**Memory: Pages per second** shows the rate of pages read from or written to disk to satisfy references to pages that are not in real memory.

**Memory: Available Bytes** is the amount of available real memory, memory that is not yet committed.

**Memory: Committed Bytes** shows the virtual memory that has been committed. Note that this metric includes pagefile space.

**Memory: Pool Nonpaged Bytes** shows the real memory in the system kernel.

**Memory: Cache Faults per second** shows the rate at which the system cannot find data in the cache and must access disk.

NT utilizes real memory, the file system cache and backing disk to implement a virtual memory management system. NT provides a 4 GB addressing range, 2 GB for the system and 2 GB for each application.

Real memory in NT is held in two categories: paged and non-paged. Paged memory is virtual memory, the virtual memory space that is mapped by NT to real memory pages and utilizes the pagefile.sys file to back pages. Non-paged real memory does not have disk backing and is used for system kernel resources which must be readily accessible at all times.

Paging in NT occurs in what are termed soft and hard page faults. Soft page faults occur when a process accesses information that is not currently in real memory and sufficient real memory is available. A hard page fault occurs when real memory is not immediately available and must be stolen from some other process to satisfy the currently running process's need for data access. Hard page faults degrade server performance rapidly.

To assess the level of memory commitment (or over commitment) on a server, the relationships between several metrics should be reviewed.

If the pages per second counter is high, a memory bottleneck is probable. Over 5 pages per second for an extended period is a generally accepted guideline, though spikes of 20 to 50 pages per second are also a good indication of memory over commitment. One exception to this guideline are servers for database applications including Domino. Large servers which are well configured may see 100 pages per second or more in normal operation.

*The primary metric for determining memory over commitment is pages per second.*

The available bytes counter will normally be low as NT will keep memory in use for disk caching. If this counter is decreasing to the NT minimum of 4GBs, the next step is to examine the disk drives that support the paging file.

Since memory is backed with disk via pagefile.sys, considerations for assessing memory also include the review of some disk metrics. Perfmon provides three counters of interest when reviewing the pagefile.sys disk(s). An average disk queue length greater than 2 on the pagefile(s) is a good indication that there is a memory bottleneck. If the %disk time metric is high and the Disk bytes per second metric is large, these counters also suggest a memory bottleneck.

*Some Disk metrics are important for assessing memory commitment levels.*

A calculation utilizing two Perfmon counters is also helpful in determining a memory shortage. These two counters are Memory:Pages per second and Logical Disk:Avg. Disk seconds per Transfer. If the product of multiplying these two counters is over 10 on a regular basis, then it is highly likely that the server would benefit from additional memory. Note that in these considerations, the Disk seconds per transfer is for the disk containing pagefile.sys and that this calculation presumes that the only activity on that disk is for pagefile.sys.

If memory is over committed and adding memory to the server is not possible, consider offloading some work. If offloading work is not possible then the server's responsiveness may be improved by spreading the paging file across two disks.

By default, NT creates one pagefile.sys during installation. To spread the paging, from the Control panel, click on the system icon, select the performance tab and at the bottom of the screen, click the **Virtual Memory** button. In the resulting screen, you can add new paging files. Only after the new files are in use, should you remove the old pagefile.sys.

*If possible, dedicate two disk drives to paging files, select the fastest disks with the lightest load.*

NT will automatically extend the size of pagefile.sys and when it does, resources are used that would not otherwise be needed. With proper allocation for pagefile.sys, dynamic paging file growth can be avoided. Placing the pagefile on it's own partition is also a good idea as this action will prevent the pagefile from getting fragmented and thus decreasing performance.

The perfmon memory object has two counters of interest when considering pagefile.sys size. These counters are Committed bytes and Commit limit. Anytime Committed bytes is greater than Commit limit, the page file has been dynamically extended. Set the paging file size to Committed bytes (monitored over a good period of time) plus 10 percent. A good place to start for the pagefile.sys size on a new server is memory size plus 10 percent.

## Disk

Disk stores both the program objects needed for execution and the associated data that these programs need to access to function. No matter how fast the processor is or how much memory is installed on the server, sooner or later the system is going to have to access disk. Since the disks are orders of magnitude slower than the processor, utilizing disk well is key to good responsiveness.

The Perfmon counters for disk must be turned on (they are off by default) using the Diskperf command as previously described in the section addressing Perfmon overhead. Remember to use the -YE option if you have RAID devices.

Perfmon provides disk objects for both logical disk and for physical disk. Physical disk counters are for individual disk drives and are the best indication for the drive as a whole. Logical disk counters provide data for logical partitions of individual physical drives or for logical partitions which span multiple physical drives as is generally the case with RAID devices. Note that if you are using hardware RAID, the RAID set appears as one disk to Perfmon.

When assessing disk bottlenecks, the following Perfmon counters for either physical or logical disks are of interest. In general, most analysis is performed using the logical object.

**% Disk Time** shows the percent of time that the disk drive is busy.

**Disk Queue Length** shows the number of disk IO requests waiting to receive service from the disk.

**Disk Bytes per Second** shows the transfer rate to and from the disk.

**Average Disk Seconds per Transfer** shows the average time to transfer data from or to the disk.

**Average Disk Bytes per Transfer** shows the average transfer size to or from the disk.

**% Disk Read Time** shows the percentage of time that the disk is busy with read requests.

**% Disk Write Time** shows the percentage of time that the disk is busy with write requests

Different performance analysis practitioners will assert different busy level guidelines for consideration when a disk is too busy, generally ranging from 50 percent to 80 percent. If a disk is not busy at all, then new work can begin almost immediately but for the disk to be justified some level of usage is to be expected.

*The point to remember is that if a disk is busy then it is not available for new work.*

Monitoring the Disk Queue length in conjunction with the % Disk Time counter is good practice. A strong indication of a disk bottleneck is when the % Disk Time counter value is high and the Disk Queue length is also high. A queue length greater than 2 for a physical disk is generally considered high.

A different approach is needed for logical disks where multiple physical drives comprise a logical drive as is common with RAID disks. In this case, the average queue length will obviously be higher. A widespread approach to evaluating queue length in this situation is to take the number of physical disks in the logical drive times 2, less 1. For example, on a logical disk that is composed of 5 physical disks, a queue length greater than 9  $((5 * 2) - 1)$  would generally be considered high.

*A more stringent approach, valid for disk intensive applications or any situation where disk subsystem responsiveness has a high priority, is to take the number of physical disks in the logical drive less 1 as the target queue length.*

A calculation utilizing two Perfmon counters is also helpful in assessing disk performance. These two counters are Disk Queue Length and Average Disk Seconds per Transfer. The product of multiplying these two counters added to one half of the Average Disk Seconds per Transfer gives the average time an IO request spends in the queue, waiting for it's chance to use the disk.

When looking at Average disk bytes per transfer, remember the higher this value, the more efficiently the server is running. Small transfer requests suffer higher overhead.

The % Disk Read and Write times can be used to calculate the read/write ratio. If this ratio is high then the disk is a good candidate for RAID 5. RAID 5 provides great fault tolerance and works well with predominately read activity. If the read/writes ratio is low then this disk is a good candidate for RAID 1 which works better than RAID 5 for predominately write activity.

*The read/write ratio is helpful when considering device characteristics such as those for RAID.*

## Network

Today's networks are complex and growing more complex. Many areas can be sources of issues for network performance. Perfmon provides a number of objects and corresponding counters that may be monitored and can help in determining the general health of the network. Other more network centric, specific tools may be required, depending on the issue.

**Server: Bytes Total per second** shows the total of the bytes both sent and received by the server over the network.

**Network Interface: Bytes Sent per second** shows the number of bytes sent using the network adapter.

**Network Interface: Bytes Total per second** shows the total number of bytes sent and received using the network adapter.

Perfmon generally organizes network counters by the transport protocol. Each has a slightly different set of counters though most counters are the same for each protocol. For TCP/IP, the following counter are of interest.

**TCP Segments per second** shows the number of TCP segments that are sent and received.

**TCP Segments Retranslates per second** shows the number of segments that are retranslated.

**UDP Datagrams per Second** shows the total number of datagrams sent and received.

**Network Interface: Output Queue Length** shows the queue depth of the output packet queue (number of packets waiting). A queue depth greater than 2 normally indicates some network congestion and is reason for investigation.

Note that the Network Interface counters and the TCP/IP counters are not available unless the SNMP service is installed. To install SNMP, go to the control panel and select Network services.

Many of the counters available for each transport protocol show error counts such as the Frame Bytes Resent per second. For most of these, the absolute value at any point in time carries little meaning. However, observation of the trend in these error counts across time can be instructive, providing a good indication of a developing problem.

## Using Perfmon with Domino Statistics

Perfmon provides the capability to acquire and display data from other programs that are reporting statistics. These other programs are normally referred to as add-in programs. Domino provides statistics that can be monitored with Perfmon using such an add-in program.

To access this facility, the Notes Performance Monitor component must be installed, either as part of the original Domino Netfinity server install or as a follow on activity. If the Notes Performance Monitor component was not originally installed, rerun the Domino installation program using the customized installation panel and select to install only the monitor.

Once the install is complete, a registration process to update the NT registry is required. The exec which does this update is *notesreg.bat* and must be executed pointing to the servers Domino directory. For example, if the server's Domino directory is D:\Lotus\Notesr4 then the command would be:

```
notesreg.bat D:\Lotus\Notesr4
```

When this exec finishes, the add-in process is complete and Domino statistics can be reported using Perfmon. Make sure that Domino is up and fully initialized before becoming concerned about the add-in setup process. Utilizing the Domino Console to issue show server and show stat commands can help with getting Domino initialized so that it shows up in the Perfmon counters.

*Note that if Domino is not started or not fully initialized, no information will show up in Perfmon.*

Domino provides many counters that can be accessed directly from the Domino console so why bother using Perfmon? The answer is two fold.

First using Perfmon provides all the facility of the Perfmon logging feature for Domino counters. Utilizing Perfmon logging is a convenient method for retaining Domino statistics and subsequently accessing the log as input to the chart or report function provides facility for reviewing the accumulated Domino metrics.

Second, having the Domino counters in Perfmon facilitates efforts to correlate systems level metrics such as the processor busy with Domino values such as the number of Domino users. The capability to observe how different system metrics and Domino metrics track can provide needed insight to the behavior of your Domino Server.

Many Domino metrics are appropriate for use in this type of correlation effort. Several that are often of interest are listed below. Of course the area of investigation will suggest the most appropriate metrics.

- Server.Trans.PerMinute
- Server.Users
- Mail.AverageDeliverTime
- Mail.AverageSizeDelivered

Any of the above Domino metrics can be correlated with system level metrics. Candidate system metrics which would normally be of interest include:

- %Processor time.
- Processor Queue Length
- Pages per second
- % Disk Time
- Disk Queue Length

## Conclusion:

The Perfmon tool provides a ready facility for acquiring, displaying and storing both system level metrics and Domino metrics for both collective analysis and for correlation. Regular use of Perfmon facilities can provide significant insight into system performance characteristics as well as bottleneck identification and resolution. Extending the effort to include Domino statistics provides a viable approach for developing a deeper understanding of the behavior of Domino Netfinity servers.