

March 2006 Online Tutorial
Written by Ryan Antony Lewis

Benchmarking Systems with Performance Monitor

Introduction

Performance Monitor is a tool that can be used to keep track of system resources, and is displayed in numerical and graphical forms. The reason that Performance Monitor is a useful security tool (as well as a system tool), is that a user can identify the average operating speeds at certain conditions and use that standard as a guide for analysis. Benchmarking consists of gathering data as a platform for later analysis, and in our case is system information on speed and performance. With the advent of new malicious software becoming harder to detect, having the system benchmarked will allow the user to run their own tests and analyze whether the system is facing performance or system problems/delays.

The clear problem with this technique is that the system must be completely clean when the initial test is conducted, or there is no way to determine what speed is regular operating level to that of an infected system. The user may try to analyze when running Windows XP in "Safe Mode", to determine how well the system runs when few services, programs, and system settings are loaded. The drawback to this technique is the great system complexity difference between a system running clean in safe mode and one that is running normally. A system will never run as well in normal boot up as in safe mode, and for this reason having the system benchmarked to safe mode settings would not be helpful in comparison to performance on a regular basis.

This is where benchmarking has its obvious drawbacks; there is always a possibility in today's information world that we have some form of infection. However this tutorial has been made for the end-user who has already determined the system to be clean. Other tutorials explain and demonstrate how to search for infections and malicious software, and the system should not be benchmarked until it has been fully tested and cleaned.

So, now that you have a clean system it makes sense to have it tested and analyzed.

Starting Performance Monitor

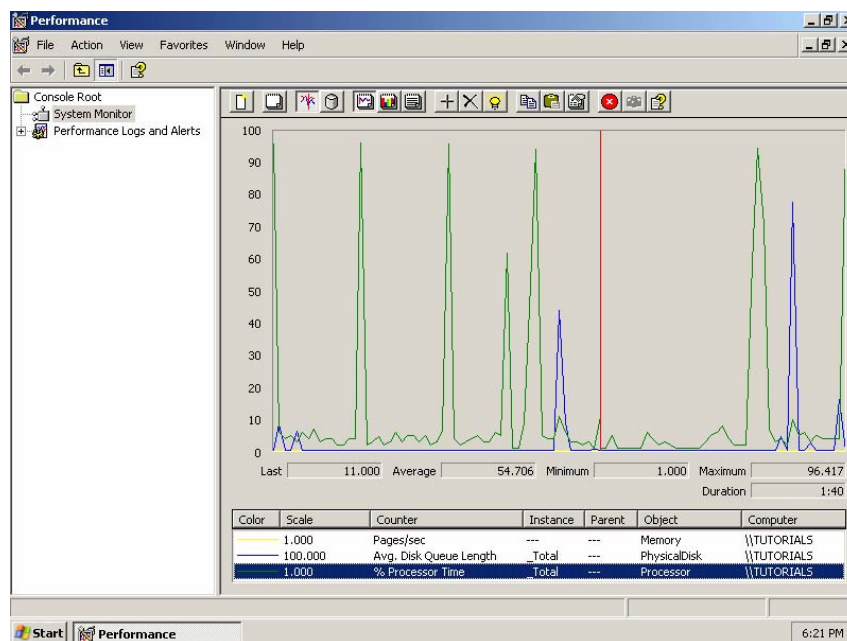
There are many ways to start Performance Monitor the easiest is using the Run command,

1. Click on Start, Run...
2. Type 'perfmon'

Or,

1. Click Start, All Programs, Administration Tools, Performance

As soon as the program loads it will begin monitoring three counters, and display them on a graph with statistics on the bottom:



The initial counters are Pages/sec, Avg. Disk Quota, and % Processor Time. These counters are a great start to show system speed and performance, however for our purposes we will need to add more counters of our own. To remove counters we hit the "X" (Delete key) button, and to add new counters we use the "+" (Ctrl-I) button. To best determine how our system is running in regards to malicious software we need to examine the most directly hindered areas of the system. Counters are separated in to "Objects" or categories. Depending upon what software has been installed, the counters will vary from each system. The following are counters that are most helpful in determining our current performance and are written as Category: Counter.

1. Memory: Pages/sec.

Microsoft Definition

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.

Ryan's Definition

When the system has to write or read data to the disk as a "page" or block of data. This helps catch faults from application problems, and shows when the memory is being used up in excessive amounts.

2. PhysicalDisk: Avg. Disk Queue Length.

Microsoft Definition

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.

Ryan's Definition

Will show how many requests to write or read from the hard drive by the system. This is helpful in showing excessive disk usage by rogue programs.

3. Processor: % Processor Time.

Microsoft Definition

% 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%.

Ryan's Definition

Shows the percentage of the CPU used during that sample. This is very important when testing to see if background processes, or rogue programs are using up excessive CPU speed.

4. System: System Cycles/sec.

Microsoft Definition

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.

Ryan's Definition

Shows combined amount of calls made to the operating system, determined by samples. This shows us how much the system is being requested.

5. IP: Datagrams/sec.

Microsoft Definition

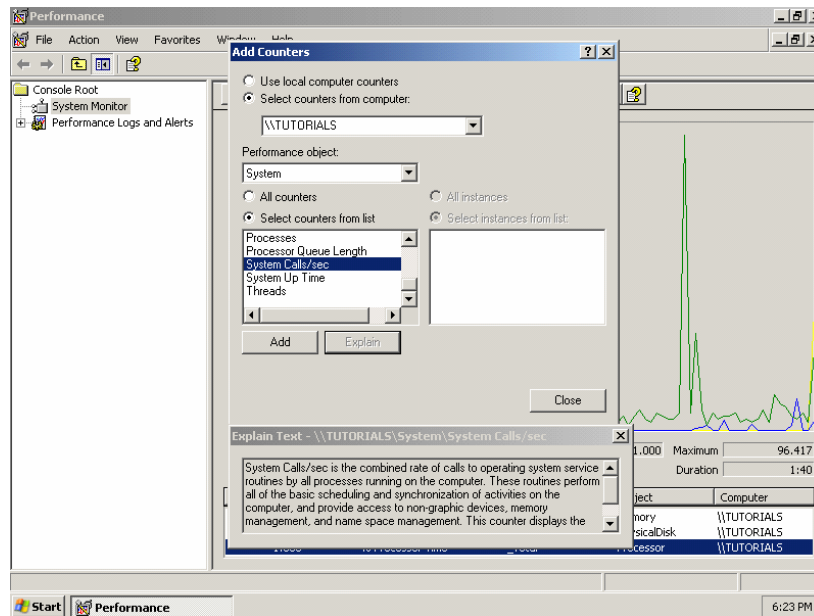
Datagrams/sec is the rate, in incidents per second, at which IP datagram's were received from or sent to the interfaces, including those in error. Forwarded datagram's are not included in this rate.

Ryan's Definition

Shows the amount of information passing through your internet connection. This is paramount in determining if internet resources are being drained by other programs during resting state.

Adding New Counters

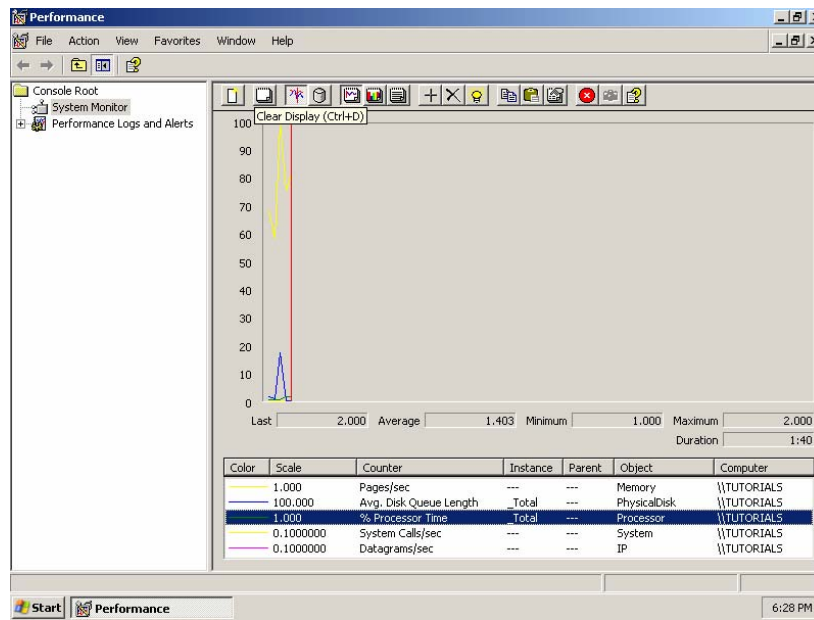
To add these two new counters to the list, we need to click on the "+" (Ctrl-I) sign on the tool menu. When the menu appears we are given a variety of choices, to begin with we will select System Cycles/sec. To add the System counter we must select "System" it from the "Performance Object" menu. Then clicking on the "Select counters from list" bubble, we may scroll down the list of System counters.



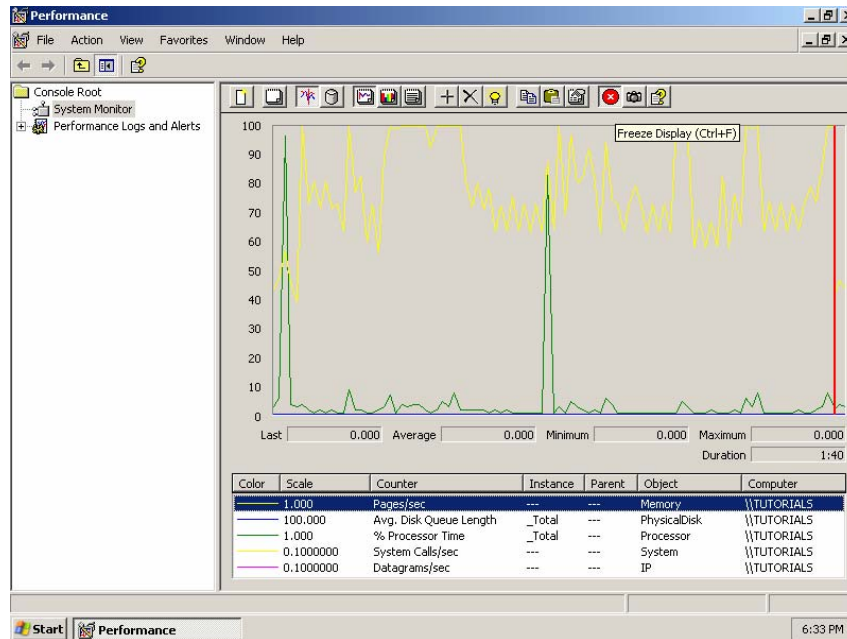
When you have found “System Calls/sec” select it by clicking the “Add” button. The counter will be added, however the menu will remain on top of the monitor. Go ahead and add the “Datagrams/sec” counter by choosing IP from the “Performance object” drop down menu. Once you have added both of the new counters, hit the “Close” button. You will now see the two extra counters on the bottom, and two new colored lines will begin to display data on the graph.

Running Performance Monitor

Now is a good time to clear the display, to start with a fresh view and can be accomplished by hitting the “Clear Display” button on the tool menu, or hitting Ctrl-D.



Now that we have all the counters necessary for testing, we should let the monitor run until the end of our “Duration” which by default is 1 minute and 40 seconds. Avoid opening any programs, running any tasks, or using the mouse, keyboard, or other peripheral device. Once the graph has almost reached the end of the duration we can freeze the display by hitting the red X sign, or Ctrl-F.

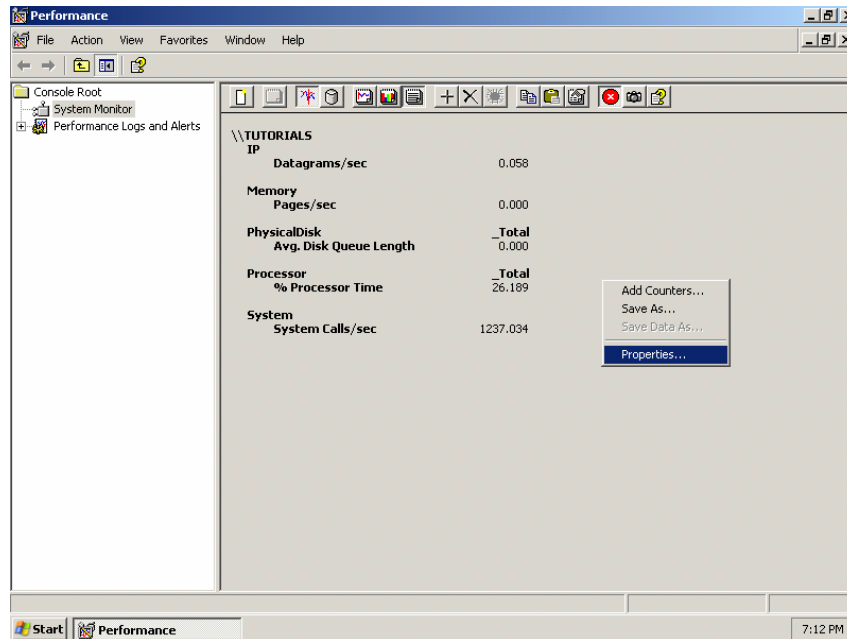


Now you can see the usage over the duration. As you can see there was much activity over the graphs, showing high peaks of usage and lower intervals. In the above picture, it may be noted that the Avg. Disk Queue Length had not received any usage time. The lack of disk write or read queues occurs as I did not save or read any data (and that includes using applications as they may be writing or reading to disk in the background), and the time spent was idle. It is very important that the computer is left idle during the testing process, as even moving the mouse pointer will cause some resource usage on the machine.

Data, Reports, and Saving

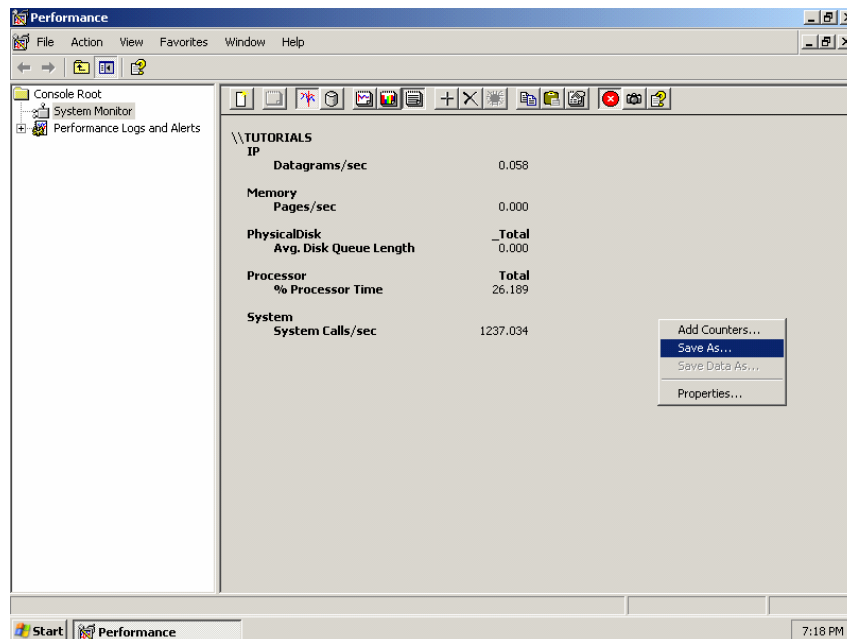
To fully utilize the benchmark process, it is best to apply more stress on the machine than you would use at an idle state. You can use a variety of techniques to stress your computer's resources, and some have been outlined in the following section. The best area to gather data for retrieval in Performance Monitor is in the Report section. You can access this area by clicking on the “View Report” button on the tool bar, or use Ctrl-R.

In the Report view you can choose which numbers to show by right clicking on the Report screen and select “Properties...” from the menu.



You need to select the “General” tab at the top, and then “Average” under the “Report and histogram Data” box. This will show averages as the numbers on the Report and allow for easier benchmarking. Now you need to store this information for later comparisons. There are many ways to keep the benchmark data, however the best way is to let the monitor do the work for you.

Right click on the Report, and click Save As.



When in the Save As menu you have a choice to save the file as either “Web Page (.HTML/HTM)” or “Report (.TSV)”. The Web Page setting is helpful, as it will save a dynamic copy of your statistics onto a HTML document. You can even take new averages and test data by opening this file and using it much like Performance Monitor on your machine. However, for benchmarking purposes the best way to use the save function is in the Report format.

The Report format will save the file as a text document, which when opened (use WordPad not Notepad) will show all the report data and other information. It is important to save this file to an area you use for security (such as “C:\BACKUP\”) or on portable media for later use. When you have saved you benchmarked information you now have basis for testing your system in the future. It is a good idea to name the benchmarks a date; as if you change hardware or software you should consider making a new benchmark. Additionally, you may use different counters for different types of tests and save them as their own files. For instance, the BST stress test mainly utilizes the Datagrams/sec test, and you could save the information purely for that counter.

Stress Tests

To get the best possible benchmark for you computer, it would be wise to try out different stress scenarios. For instance, when you load up Windows and login and only the initial startup programs have begun, do not start any applications or services and just run Performance Monitor. Another scenario may be. Below are possible scenarios in more detail:

Browser Stress Test (BST)

A browser stress test will allow the “Datagrams/sec” counter (which was also left blank, due to lack of usage in previous screen shots) to test internet usage through your browser. Basically, it is searching the internet, and monitoring how much data is sent back and forth during that time. During this test, many of the other five counters will be stressed; however it is mainly used to stress the “Datagrams/sec” test.

Step 1

When you start a BST you should make sure that you Temporary Internet Files are cleared if using Internet Explorer or the Cache if you are using Mozilla Firefox. Additionally clean any cookies and history, to make sure that you are entering these pages as if for the first time. This will make sure that you use some bandwidth and receive files. To accomplish any of these tasks please refer to your individual browser help file.

Step 2

Now you need to have previously determined sites to visit, preferably with higher content length. Some examples are:

- + <http://www.ryanalewis.com>
- + <http://www.msn.com>
- + <http://www.cnn.com>
- + <http://www.amazon.com>

Additionally you should pick a site that has a downloadable file on their site, so that you can see how your usage fluctuates on the same file and at the same website (unfortunately not the same time, which is a factor we cannot avoid).

- + <http://www.microsoft.com/downloads/search.aspx?displaylang=en&categoryid=2>
Direct-X Download, as of March 06.
- + <http://www.ryanalewis.com/TutorialData/largefile2.dat>
A file I have sitting on my site for you to use.

Step 3

Start the Performance Monitor, making sure all the counters needed have been selected. Clear the screen, and make sure the Report view is on, and you have averages selected.

Step 3

Use the pre-determined sites in the same order, many times over until you have a good average. Remember that you need to remove your temporary files before every attempt at the test, and you should delete the large file each new test. Once you have a confident value for the "Datagrams/sec" you can stop the test.

Machine Stress Test (MST)

A machine stress test will show usage on all other counters than "Datagrams/sec". This test will make use of the processor, memory, disk queue, and system calls. To test all these counters is actually very simple, as ordinary computer usage will use these areas, however adding particular stress can give better results when comparing to an infected (and considerably drained) system.

Step 1

Download the provided large file sitting on my server, or download another from other sites. We will use this file to write to our disk. Do not begin testing until you have already downloaded this file to your disk; the best time is after conducting your BST.

+ <http://www.ryanalewis.com/TutorialData/largefile.dat>
The large file that you use for copying.

Step 2

Start the Performance Monitor, making sure all the counters needed have been selected. Clear the screen, and make sure the Report view is on, and you have averages selected.

Step 3

Now you need to copy your file many times over, to make use of all your system resources that we are currently monitoring. My own script address is below, which will copy the large file from my website over and over. To use my script properly you must put my file in the same place as the script file. You can write your own scripts, or even use your own even large file, however copying at fast rates will get the "Avg. Disk Queue Length" counter to spike. You will require 35MB to run the test once, and the room grows exponentially from there. If you find your computer is running the test too fast, simply run the MST.bat file again and it will double the strain on your disk.

+ <http://www.ryanalewis.com/TutorialData/MST.bat>
The BAT script that will copy the large file repeatedly.

NOTE: Users of some antiviral/anti-malware programs will have to allow the use of batch scripts, and may be prompted for permission to run the "MST.bat" script. To use this script, you must allow the program to run.

Once the script (or your own) has stopped running, stop the counter. If your computer cannot run the script in under a minute forty seconds, you may need to use your own script as the data will be written over.

Analysis and Advanced Information

The benchmarks are utilized when testing a possibly infected system, by testing that system and comparing the new information with the old. If there is a significant increase or change in resource usage, then the user can be satisfied that either there is indeed an infection, or that the computers performance has dwindled. This technique is not a proof positive; however it is useful in determining if the computer is indeed running slower than previous times.

It must be noted that most systems will degrade over time due to memory use and applications. Benchmarks should be re-evaluated on a regular basis, and are only useful if used as a possible indicator to an infection or performance issue. When comparing statistics it is very important that the necessary controls of the experiment were in place, so that only the manipulated variables have a direct impact on the statistics.

The BST and MST tests are to be used in a controlled environment and should not be abused. These tests will give a good indicator of how your system runs under stress, and will give a good basis for benchmarking your system. There are many other ways to test your systems, and it is important to use regular checks on your performance level.