

|  |
| --- |
|  |
| Performance Monitoring Tool - Perfmon |
|  |

Kolla Madhavilatha

June 7, 2017

Madhavilatha.kolla@cognizant.timeinc.com

Table of Contents

[1. The Main Reasons for Performance Monitoring 2](#_Toc484758894)

[2. Perfmon: Introduction 2](#_Toc484758895)

[3. Installation 3](#_Toc484758896)

[4. Invoking Perfmon 3](#_Toc484758897)

[5. Basic Monitoring 4](#_Toc484758898)

[6. Logging Monitoring (Automatic) 4](#_Toc484758899)

[7. Perfmon Counters 4](#_Toc484758900)

[8. Perfmon Integration to Load runner 6](#_Toc484758901)

[9. Perfmon Integration to Apache Jmeter 8](#_Toc484758902)

# The Main Reasons for Performance Monitoring

Performance Monitoring is examining how programs are running on their computers and affect the computer's performance.

Below are the following reasons –

* Detecting network bottlenecks.
* Troubleshooting latency frustrations.
* Identifying server performance problems.
* Uncovering intermittent faults.
* Planning the capacity of your servers and subnets.
* Setting alerts so that you can nip trouble in the bud.
* Creating baselines when activity is low.
* Understanding the effect of your workload on resources.

# Perfmon: Introduction

**Performance Monitor** or **PerfMon**for short can be used to monitor real time performance of the system, capture various metrics and gives the freedom to choose the counters are in monitoring scope for longer duration.

It captures information about the hardware, operating system, SQL Server and more. The whole process is automated and so is the data collection.

Performance Monitor (Perfmon) is a powerful program found in all Microsoft servers from NT to Windows Server 2008 having capability to track the system performance –

* Disk
* Memory
* CPU
* Network

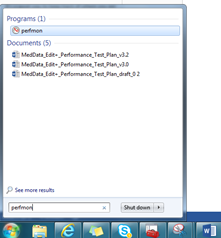
**PerfMon**can be used to collect OS and hardware resource counters as well as SQL Server counters and captures real-time data to be viewed and analyze in multiple ways.

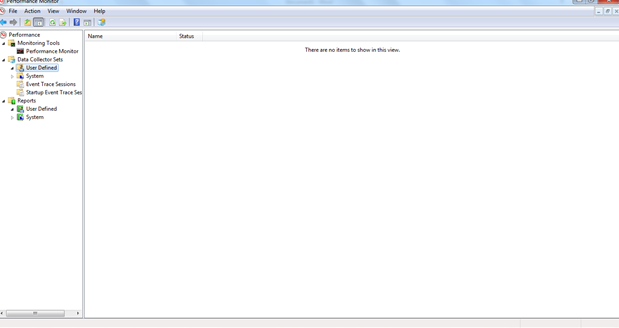
# Installation

On Windows 7, this is a free tool that you do not need to download.

# Invoking Perfmon

Press Windows -> Type “Perfmon” without quotes.



Perfmon GUI –

# Basic Monitoring

Basic monitoring helps to monitor the computer short period of time only.

* Go to Monitoring tools > Performance Monitor.

By default, it will display example graph that shows **"%processor time"** or CPU usage.

To add additional graphs, press **Ctrl + N** or click green plus button in the upper toolbar.

Select the required counters from left pane and click **Add -> The list will populate in the right pane and OK.**

# Logging Monitoring (Automatic)

Automatic monitoring doesn’t need to interact with performance monitor and it saves report as text file.

* Go to **Data Collector sets > User Defined > Right click at right pane > New > Data Collector set**
* In the window type in name of collector (anything you want) and select type and click next.
* Select folder where you want log files to be stored. Click next.
* Select **Open properties for this data collector set.** Click finished.
* A window will appear. Go to schedule tab. Click Add to add schedule.
* To Start collecting the data, go to **Data Collector sets > User Defined > right click at your collector set >Start.**
* To see reports, go to **Reports > User Defined > [name] > select report.**

**Note**: Make sure that the data management and report generation option is enabled under reports column: (Click on properties > check on enable data management and report generation > click on apply > click ok)

# Perfmon Counters

* **Logical Disk\% Free Space**: This measures the percentage of free space on the selected logical disk drive. Take note if this falls below 15 percent, as you risk running out of free space for the OS to store critical files. One obvious solution here is to add more disk space.
* **Physical Disk\% Idle Time:**  This measures the percentage of time the disk was idle during the sample interval. If this counter falls below 20 percent, the disk system is saturated. You may consider replacing the current disk system with a faster disk system.
* **Physical Disk\Avg. Disk Sec/Read:**  This measures the average time, in seconds, to read data from the disk. If the number is larger than 25 milliseconds (ms), that means the disk system is experiencing latency when reading from the disk. For mission-critical servers hosting SQL Server and Exchange Server, the acceptable threshold is much lower; approximately 10 Ms. The most logical solution here is to replace the current disk system with a faster disk system.
* **Physical Disk\Avg. Disk Sec/Write:** This measure the average time, in seconds, it takes to write data to the disk. If the number is larger than 25 ms, the disk system experiences latency when writing to the disk. For mission-critical servers hosting SQL Server and Exchange Server, the acceptable threshold is much lower; approximately 10 ms. the likely solution here is to replace the disk system with a faster disk system.
* **Physical Disk\Avg. Disk Queue Length:**  This indicates how many I/O operations are waiting for the hard drive to become available. If the value here is larger than the two times the number of spindles, that means the disk itself may be the bottleneck.
* **Memory\Cache Bytes:** This indicates the amount of memory being used for the file system cache. There may be a disk bottleneck if this value is greater than 300MB.
* **Memory\% Committed Bytes in Use:** This measures the ratio of Committed Bytes to the Commit Limit—in other words, the amount of virtual memory in use. This indicates insufficient memory if the number is greater than 80 percent. The obvious solution for this is to add more memory.
* **Memory\Available Mbytes:**  This measures the amount of physical memory, in megabytes, available for running processes. If this value is less than 5 percent of the total physical RAM, that means there is insufficient memory, and that can increase paging activity. To resolve this problem, you should simply add more memory.
* **Memory\Free System Page Table Entries:** This indicates the number of page table entries not currently in use by the system. If the number is less than 5,000, there may well be a memory leak.
* **Memory\Pool Non-Paged Bytes:** This measures the size, in bytes, of the non-paged pool. This is an area of system memory for objects that cannot be written to disk but instead must remain in physical memory as long as they are allocated. There is a possible memory leak if the value is greater than 175MB (or 100MB with the /3GB switch). A typical Event ID 2019 is recorded in the system event log.
* **Memory\Pool Paged Bytes:**  This measures the size, in bytes, of the paged pool. This is an area of system memory used for objects that can be written to disk when they are not being used. There may be a memory leak if this value is greater than 250MB (or 170MB with the /3GB switch). A typical Event ID 2020 is recorded in the system event log.
* **Memory\Pages per Second:**  This measures the rate at which pages are read from or written to disk to resolve hard page faults. If the value is greater than 1,000, as a result of excessive paging, there may be a memory leak.
* **Processor\% Processor Time:** This measure the percentage of elapsed time the processor spends executing a non-idle thread. If the percentage is greater than 85 percent, the processor is overwhelmed and the server may require a faster processor.
* **Processor\% User Time:**  This measures the percentage of elapsed time the processor spends in user mode. If this value is high, the server is busy with the application. One possible solution here is to optimize the application that is using up the processor resources.
* **Processor\% Interrupt Time:**  This measure the time the processor spends receiving and servicing hardware interruptions during specific sample intervals. This counter indicates a possible hardware issue if the value is greater than 15 percent.
* **System\Processor Queue Length:**  This indicates the number of threads in the processor queue. The server doesn’t have enough processor power if the value is more than two times the number of CPUs for an extended period of time.
* **Network Interface\Bytes Total/Sec:**  This measures the rate at which bytes are sent and received over each network adapter, including framing characters. The network is saturated if you discover that more than 70 percent of the interface is consumed. For a 100-Mbps NIC, the interface consumed is 8.7MB/sec (100Mbps = 100000kbps = 12.5MB/sec\* 70 percent). In a situation like this, you may want to add a faster network card or segment the network.
* **Network Interface\Output Queue Length:** This measures the length of the output packet queue, in packets. There is network saturation if the value is more than 2. You can address this problem by adding a faster network card or segmenting the network.
* **Process\Handle Count:**  This measures the total number of handles that are currently open by a process. This counter indicates a possible handle leak if the number is greater than 10,000.
* **Process\Thread Count:** This measures the number of threads currently active in a process. There may be a thread leak if this number is more than 500 between the minimum and maximum number of threads.
* **Process\Private Bytes:** This indicates the amount of memory that this process has allocated that cannot be shared with other processes. If the value is greater than 250 between the minimum and maximum number of threads, there may be a memory leak.

# Perfmon Integration to Load runner

Following two main steps are involved in configuring Windows Resource monitors in Controller,

* Adding the monitored server machine
* Adding Resource Measurement
* **Adding Monitored Server Machine**

Following steps are involved in adding monitored machine,

* Open LoadRunner Controller
* Select System Resource Graphs
* Double click on Windows Resources
* Right click on Windows Resource Graph area
* Select “Add Measurements”
* Add the Monitored Server machine
* Click on Add button
* Enter machine name or IP
* Select its platform
* Click on “OK” button

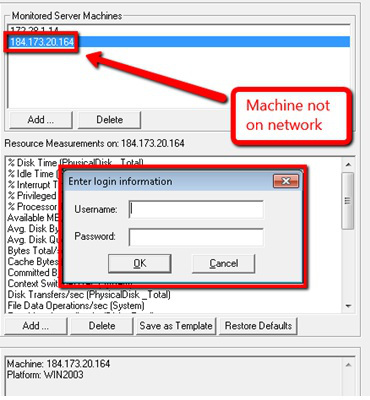
Monitored machine should be added and displayed in “Monitored Server Machine” window.

* **Adding Resource Measurement**

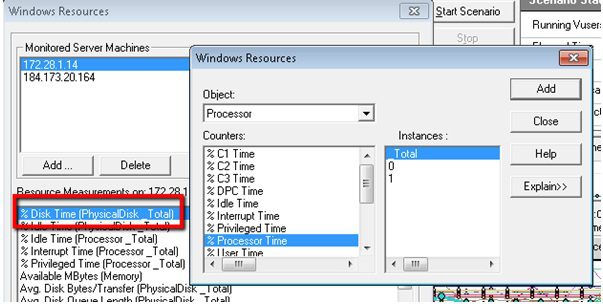
Once added the monitored machine, then need to follow below steps to add its monitors whose behavior want to check during the load test.

Click on “Add…” button to add all the above Resource Measurements

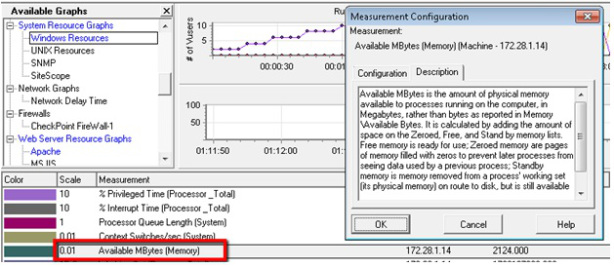
It will ask for the Username and Password if monitored system is not on the same network

[](https://sajidmanzoor.files.wordpress.com/2013/04/first-image.jpg)

Additional details of any Resource Measurement can be done by double clicking on it and selecting its component

[](https://sajidmanzoor.files.wordpress.com/2013/04/second-image.jpg)

Click on “OK” button once all the required resources are configured.

[](https://sajidmanzoor.files.wordpress.com/2013/04/third-image.jpg)

# Perfmon Integration to Apache Jmeter

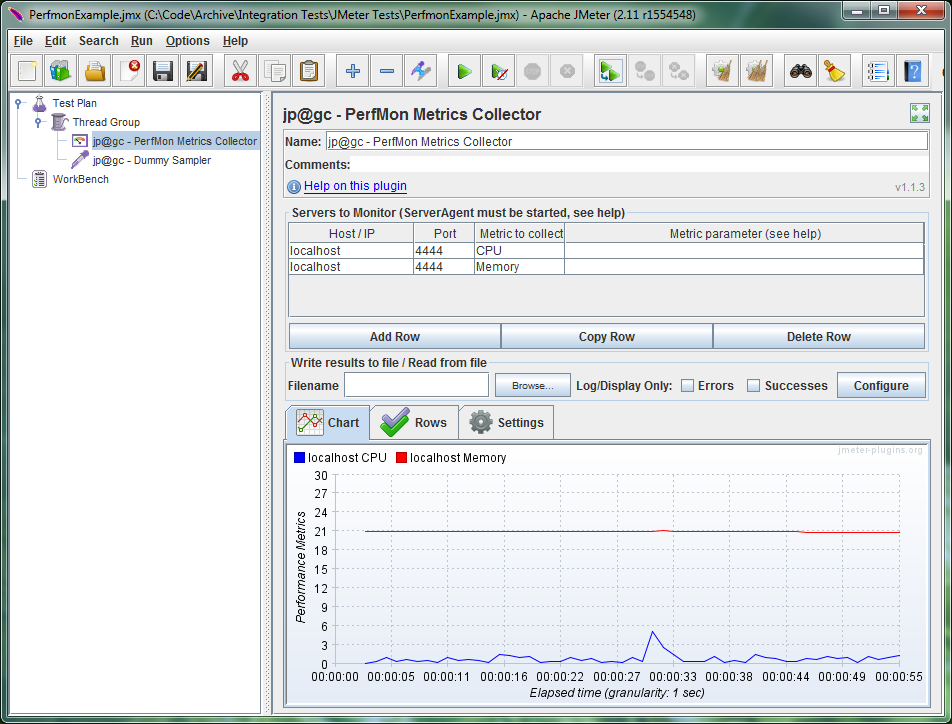
Perfmon can be integrated to Apache jmeter as a plugin available in the URL for download, which helps to get the CPU or Memory usage of a computer or server in graphical format.

URL: <https://jmeter-plugins.org/wiki/PerfMon/>

Download the “PerfMon Agent” which is available as a Zip file from the above link.

Start the PerfMon Agent by opening the “startAgent.bat” (or “startAgent.sh”) file that is included in the zip file. You need to run this on all servers you want to collect data from.

Once the PerfMon Agent is running, collect data using the PerMon Metrics Collector plugin.



Use “Add Row” to create lines on the graph, using a row per server/metric. The above example shows the process of collecting the CPU data from localhost and the Memory data from localhost.

Use the “Rows” tab to filter the collected data wish to see, for example, just the CPU data on its own.

The available PerfMon metrics include:

* CPU
* Memory
* Swap
* Disks I/O
* Network I/O
* TCP
* JMX
* EXEC
* TAIL