**What is the (SQL) Performance Monitor?**

Windows Performance Monitor or PerfMon is a great tool to capture metrics for your entire Windows server and including your SQL Server environment. It has easy to view graphs and counters that you can select to find a specific issue at hand.

There are counters for .NET, Disks, Memory, Processors, Network, and many for SQL Server related to each instance of SQL Server that you may have on the box.

The purpose of the perfmon is to identify the metrics you like to use to measure SQL Server performance and collecting them over time giving you a quick and easy way to identify SQL Server problems, as well as capture your performance trend over time.

**What are counters, what counters should we use?**

**Depends!**

Counters are a method to measure current performance, as well as performance over time. As there are many counters and there is no strict guideline as to use a specific counter, I suggest we look at some common ones for our demonstration and then allow the DBA to experience with others. The best approach is to capture some common values when your system is running fine, so you can create a baseline.  Then once you have information about these counters and the baseline you can compare performance issues against the baseline.

Measuring Windows counters example:

* Memory – Available MBytes
* Paging File – % Usage
* Physical Disk – Avg. Disk sec/Read
* Physical Disk – Avg. Disk sec/Write
* Physical Disk – Disk Reads/sec
* Physical Disk – Disk Writes/sec
* Processor – % Processor Time
* Network

Measuring SQL Server counters:

**SQLServer: Buffer Manager: Buffer cache hit ratio**

The buffer cache hit ratio counter represents how often SQL Server is able to find data pages (smallest unit of data 8k) in its buffer cache when a query needs a data page. We want this number to be as close to a 100 as possible, which means that the SQL Server was able to get data for queries out of memory instead of reading from disk. A low number indicates memory issue.

**SQLServer: Buffer Manager: Page life expectancy**

The page life expectancy counter measures how long pages stay in the buffer cache in seconds. The longer a page stays in memory, the better as this indicates that the SQL Serve did not need to read from the disk. A lower number indicates that the it’s reading form disk. Update memory??

**SQLServer: SQL Statistics: Batch Requests/Sec**

Batch Requests/Sec measures the number of batches SQL Server is receiving per second. This information provides how much activity is being processed by your SQL Server box. The higher the number, the more queries are being executed on your box. Higher number indicates that the server is busy. But, again create a baseline for you system.

**SQLServer: SQL Statistics: SQL Compilations/Sec**

The SQL Compilations/Sec measure the number of times SQL Server compiles an execution plan per second. Compiling an execution plan is a resource-intensive operation. One compile per every 10 batch requests.

**SQLServer: Locks: Lock Waits / Sec: \_Total**

In order for SQL Server to manage concurrent users on the system, SQL Server needs to lock resources from time to time. The lock waits per second counter tracks the number of times per second that SQL Server is not able to retain a lock right away for a resource. Ideally you don't want any request to wait for a lock. Therefore you want to keep this counter at zero, or close to zero at all times.

**SQLServer: Access Methods: Page Splits / Sec**

This counter measures the number of times SQL Server had to split a page when updating or inserting data per second. Page splits are expensive, and cause your table to perform more poorly due to fragmentation. Therefore, the fewer page splits you have the better your system will perform. Ideally this counter should be less than 20% of the batch requests per second.

**SQLServer: General Statistics: User Connections**

The user connections counter identifies the number of different users that are connected to SQL Server at the time the sample was taken. This does not refer to the end users, but rather connections that a user has to the server. Thus a user can have many connections

**SQLServer: General Statistic: Processes Block**

The processes blocked counter identifies the number of blocked processes. When one process is blocking another process, the blocked process cannot move forward with its execution plan until the resource that is causing it to wait is freed up. Ideally you don't want to see any blocked processes. When processes are being blocked you should investigate.

**SQLServer: Buffer Manager: Checkpoint Pages / Sec**

The checkpoint pages per second counter measures the number of pages written to disk by a checkpoint operation. If this counter is climbing, it might mean you are running into memory pressures that are causing dirty pages to be flushed to disk more frequently than normal.