

Get a Handle on Windows Performance Analysis

Michael Morales

(Reprinted from WindowsItPro Magazine)

Executive Summary

Analyzing Windows Performance Monitor logs is a time-consuming process, often requiring expert knowledge to interpret the log data for use in Microsoft Windows system performance troubleshooting. The good news is that you can simplify the process of collecting and analyzing Performance Monitor data by using three handy tools used by Microsoft support professionals: perfwiz.exe, logman.exe, and Performance Analysis of Logs (PAL).

As an administrator, you probably know firsthand that manually analyzing Performance Monitor logs is a time-consuming process, requiring both a deep-level knowledge of performance counters and familiarity with Windows architecture. The Microsoft Global Escalation Services team gets thousands of calls from customers requesting help in analyzing Performance Monitor logs either to help solve an existing problem or determine whether their servers are optimized for the best possible performance. I'll tell you about some tools that Microsoft support uses for system performance analysis, which can help you more easily and effectively analyze your systems' performance and troubleshoot performance issues.

Two-Part Analysis

Performance analysis comprises two main tasks:

1. **Data collection:** Collecting Performance Monitor logs locally or remotely can be challenging as the number of systems to be monitored increases. Another challenge is determining which set of performance counters to collect and at what intervals.
2. **Data analysis:** Once the data is collected, analyzing it correctly is yet another challenge, and it's in the analysis portion of the process that Microsoft support receives the most calls for assistance.

Let's look at tools you can use to handle each performance-analysis task.

Data-Collection Tools

Performance Monitor Wizard (perfwiz.exe) walks you through creating local and remote Performance Monitor logs. PerfWiz simplifies the process of gathering Performance Monitor logs on Windows Server 2003, Windows XP (x86 only), and Windows 2000 systems by configuring the correct counters to collect and suggesting the most appropriate sample intervals and log-file sizes. You can download PerfWiz [here](#).

Logman.exe is a built-in Windows command-line tool that manages and schedules performance counter collections on local and remote systems and runs on Windows Server 2008, Windows Vista (x86 and x64), Windows 2003, and XP. You can find information about Logman syntax and examples of usage [here](#).

Here's an example of how you'd use logman.exe from the command line to generate a Performance Monitor log called High-CPU-Perf-Log capturing data at 5-second intervals:

```
Logman.exe create counter High-CPU-Perf-Log
-f bincirc -v mmddhhmm -max 250
-c "\LogicalDisk(*)\*" "\Memory\*"
  "\Network Interface(*)\*" "\Paging File(*)\*"
  "\PhysicalDisk(*)\*" "\Process(*)\*"
  "\Redirector\*" "\Server\*" "\System\*"
  "\Thread(*)\*" -si 00:00:05
```

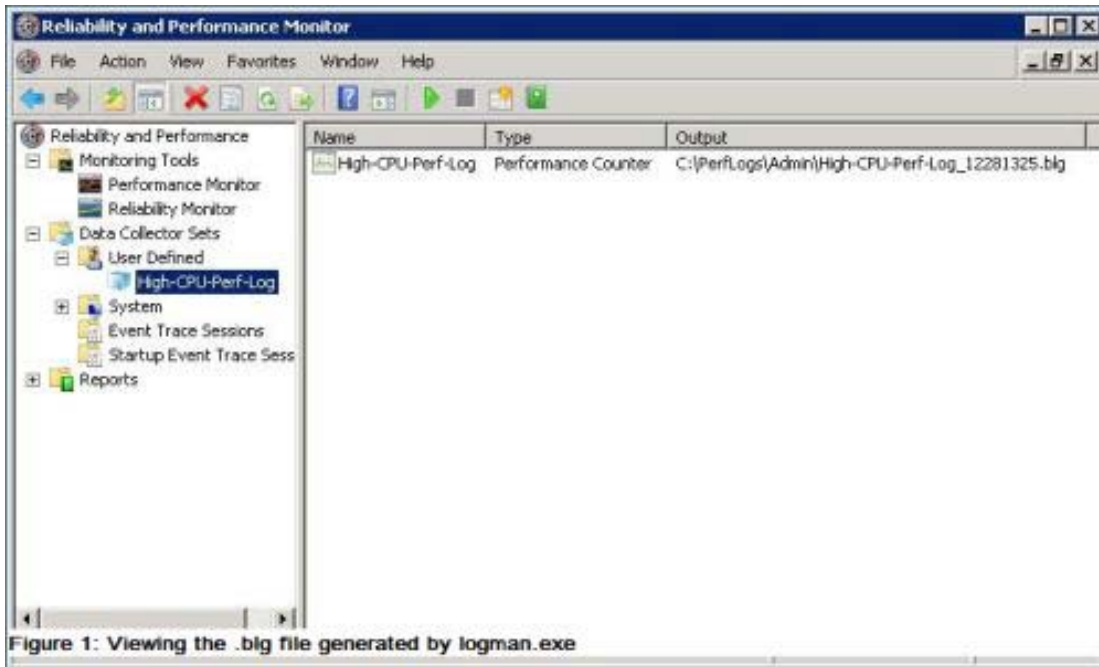
Get a Handle on Windows Performance Analysis

Michael Morales

(Reprinted from WindowsItPro Magazine)

(Be sure to type the entire command without the line breaks shown in the example here.)

Executing the logman.exe command generates a .blg file. To view this file, open Performance Monitor and navigate to the Data Collector Sets folder, as Figure 1 shows.



When you're running Performance Monitor to try to pinpoint the source of a problem on a system, I recommend you reboot the server before starting a Performance Monitor log and let the log run from reboot until the condition you're tracking occurs (e.g., system hang, sluggishness). By allowing the Performance Monitor log to collect data until the problem returns, you're ensuring that the log captures all relevant data about your problem. Gathering as much data as possible in the log will improve the reliability of the analysis. Rebooting first, then starting the Performance Monitor log helps in identifying trouble spots because you can see the resource consumption (memory, CPU, and disk) from the very beginning all the way through the problem period.

You should regularly collect baseline performance data on your most essential servers. Doing so could save you a lot of pain and effort in proving or disproving an issue exists. Often at Microsoft support, customers ask us whether a certain performance statistic is "bad" or "good." Although Microsoft provides performance guidelines, it's difficult to know what's typical for your environment unless you have baseline performance data. Creating baselines also helps you to focus on the problem at hand when there are other problems on the system. Certain performance statistics can be falsely blamed as the culprit for a new issue, but having a baseline from before the new problem occurred will help keep the focus off irrelevant statistics. (You can find Microsoft guidelines for gathering baseline data here.)

Data Analysis

A tool that Microsoft support relies on to analyze Performance Monitor logs is the Performance Analysis of Logs (PAL) Tool. Clint Huffman, a Microsoft senior premier field engineer, wrote the 6,000-line VBScript tool, which is free and open source. PAL lets administrators easily analyze Performance Monitor logs without requiring them to be experts in performance counters or Windows architecture.

Get a Handle on Windows Performance Analysis

Michael Morales

(Reprinted from WindowsItPro Magazine)

PAL contains a wizard-based UI that asks specific information about the system, which PAL passes as arguments to the VBScript program. PAL picks up where other log analyzers leave off, such as taking into account whether the system is 64-bit or 32-bit, whether the /3GB switch is used, and how much physical memory is installed—all variables that affect system performance. PAL uses these variables along with known thresholds, which were determined by engineers with years of experience, to determine the analysis that's displayed. PAL provides a chronological order of alerts, so that you can correlate your system's performance to any problems that you noticed at specific times.

PAL also can provide application-specific analysis for applications such as Microsoft BizTalk Server, Microsoft Exchange Server, Microsoft Office SharePoint Server, Microsoft SQL Server, and Microsoft IIS. So as an administrator wearing several hats, you can have application-specific performance data analyzed without being an expert in the performance counters for an application. PAL can make your life easier by providing analysis for baseline data when performance is typical or to help pinpoint the root cause of a performance issue when a problem occurs.

PAL's user-friendly UI walks you through the few steps necessary to start the analysis process. The analysis report that PAL generates is an .html file that's stored by default under the My Documents\PAL Reports folder. The report contains hyperlinks and graphs that enable easy interpretation and navigation, and the file's portability lets you easily store it in a convenient location.

Using PAL

An example of where PAL saved the day came when a customer received a Microsoft Operations Manager (MOM) alert that detected that all BizTalk services had gone offline on one of the customer's BizTalk servers. Remotely connecting to the server wasn't possible then. However, since the server was part of a cluster, the event logs were replicated, revealing the following error at the time the BizTalk server went offline:

Event Type: Error
Event Source: Srv
Event Category: None
Event ID: 2019

Description: The server was unable to allocate from the system non-paged pool because the pool was empty.

The error explained why remote connections to the server weren't possible but gave us no information about the problem's root cause.

The customer solved the immediate problem by rebooting the server, which then resumed participating in BizTalk transactions. But we needed to understand what caused the failure. To do so, our first step after the reboot was to capture a Performance Monitor log. After capturing the performance data, we copied the .blg file to our workstation for analysis, then loaded the .blg file into the PAL wizard to generate an analysis report. Using the event log error indicating that the system was depleted of non-paged pool memory, we scrolled down to view the alerts in the report's memory section and noticed 31 Handle Leak Detection alerts, as Figure 2 shows.

Get a Handle on Windows Performance Analysis

Michael Morales

(Reprinted from WindowsItPro Magazine)



Clicking the Handle Leak Detection hyperlink displayed an easy-to-read graph, which revealed that more than 340,000 handles were consumed by a single process. Additionally, we clicked the Chronological Order link near the top of the report (not shown in Figure 2), which displayed an explanation of the alert, as Figure 3 shows.

26/08/2008 1:27:35 PM	Condition	Counter	Min	Avg	Max	Hourly Trend
	Addressed Handle Leaks Suspected - more than 10,000 handles and a trend of more than 100 handles per hour	\\COMPUTER_NAME\Process\PROCESS_NAME\Handle Count	343,683	343,701	343,722	6,240,000000002

Figure 3: Handle-leak alert explanation

The process that PAL alerted us to turned out to be the problem with this particular server after we updated the process (i.e., we checked with the vendor for an update, found an update that the customer had not applied, and applied the update—which solved the problem). Bingo! No more leaking handles, and no more alerts from MOM indicating that BizTalk processes are offline.

Performance Analysis Insight

These tools and the guidance I've provided for using them should give you some insight into managing the process of system-performance analysis. This knowledge can help you better understand how your systems typically perform and, if you do encounter a performance problem, help you to resolve it with free and easy-to-use tools.