

3 Steps to Troubleshooting Device Drivers

Steven Daugherty

(Reprinted From WindowsItPro Magazine)

Executive Summary

If your applications are loading slowly or don't load at all, the culprit might be a device driver and the deferred procedure calls (DPCs) it makes. By using such tools as Windows Task Manager, Performance Monitor, Process Explorer, the Kernel Profiling Tool, and the registry, you can quickly recognize, diagnose, and solve driver performance problems.

I'm waiting for an application to start...again. As I stare at the hourglass on my Windows XP desktop, I wonder what is slowing things down this time. I press Ctrl+Shift+Esc to bring up Windows Task Manager and click the Processes tab. I'm ready to kill the offending process, but I find nothing but the usual System Idle Process at the top of the list.

Over time, Windows system performance tends to degrade. Many performance problems can be resolved by eliminating unnecessary programs from startup, defragmenting the hard disk, and cleaning the registry. In my case, however, none of these actions worked. I don't usually get into desktop support issues but since it was my desktop, I was determined to find and resolve the problem.

When I searched the Internet, I found many people complaining about the same symptoms in blogs and forums, but they didn't provide any solutions. I got a few clues when some people mentioned that the problem went away after they upgraded a driver or replaced a disk controller. At this point, I concluded that the performance problem was probably related to a driver, but I didn't know for sure. So, I dug deeper. After wading through some device-driver development articles, I found a gem. The Microsoft TechNet article "Advanced DPCs" discusses how to diagnose driver performance problems by examining deferred procedure call (DPC) queue activity.

DPCs wreaking havoc on Windows system performance is a common problem in not only workstations but also servers. Knowing how to recognize, diagnose, and solve a DPC problem can save you hours of troubleshooting and possibly save you from having to reinstall or reconfigure the problematic workstation's or server's OS.

How to Recognize the Problem

To recognize the problem, you need to know about DPCs and how to perform an initial check in Windows Task Manager when you're experiencing a performance problem. DPCs are a part of the Windows interrupt handling architecture. Interrupt handling consists of two components, both of which are part of a device driver. The first component is the Interrupt Service Routine (ISR), which quickly allows the hardware to get the device to stop interrupting. The driver actually handles the processing around the interrupt later by queuing a worker thread from the created DPC queue object. In short, this means that a driver does its initial work with a hardware device in a DPC. Because both the ISR and DPCs are vendor-provided within the driver for a device, a poorly written kernel-mode driver can have a significant impact on overall system performance.

When you're experiencing a performance problem, you can use Windows Task Manager to see whether it might be due to DPCs. When a performance problem might be related to DPCs, you'll see that the CPU usage is high on the Performance tab, but no processes are taking the blame on the

3 Steps to Troubleshooting Device Drivers

Steven Daugherty

(Reprinted From WindowsItPro Magazine)

Processes tab. You'll also see that System Idle Process is showing the highest CPU usage on the Processes tab, which is typical for an idle system.

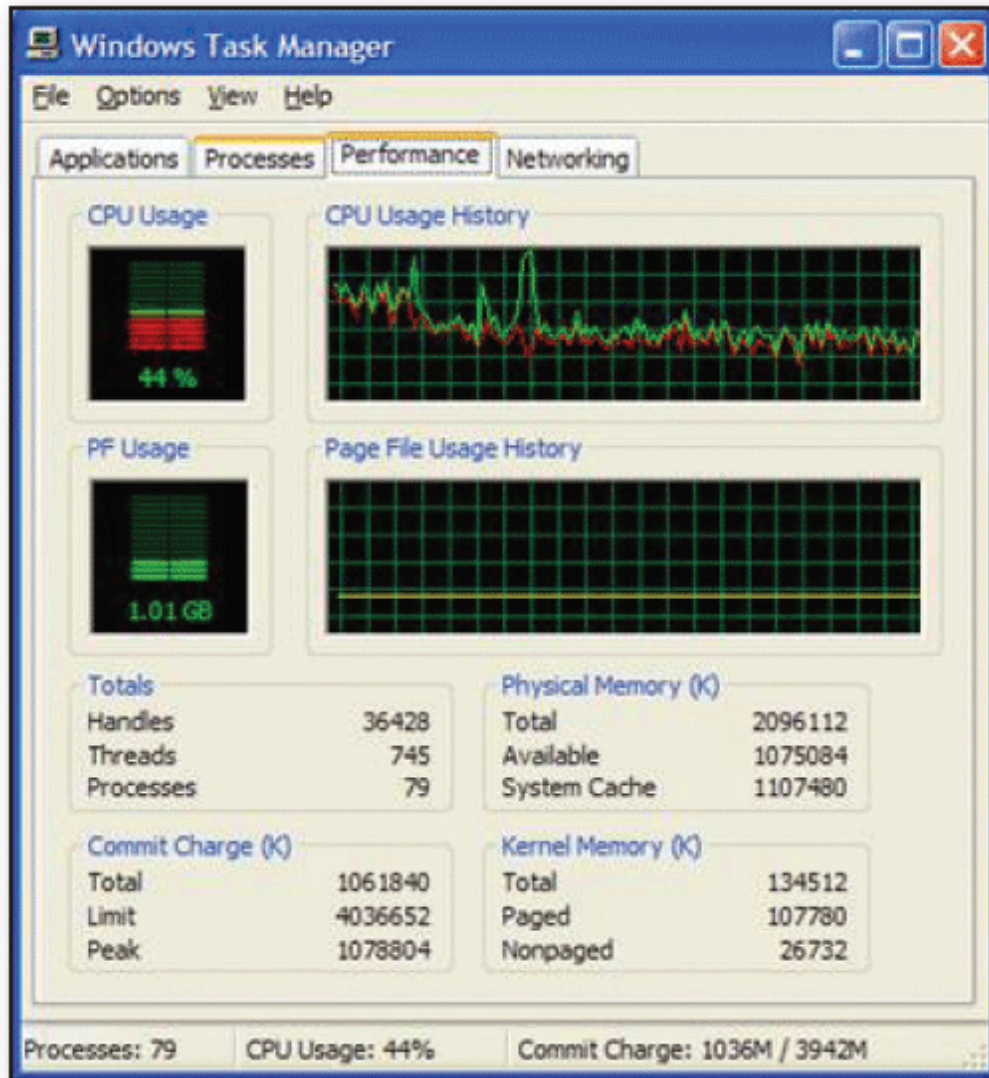


Figure 1: Using Windows Task Manager to determine how much CPU time is being used by the Windows kernel

The System Idle Process typically doesn't significantly affect system performance. However, when DPC problems exist, the kernel will be using a large percentage of the CPU, which can impact system performance. You can find out the kernel's CPU usage by selecting the Performance tab and choosing the Show Kernel Times option on the View menu. The bottom red portion of the CPU Usage graph shows the CPU time being used by the Windows kernel. The sample CPU Usage graph in Figure 1 reveals that something loaded by the kernel is monopolizing the CPU.

3 Steps to Troubleshooting Device Drivers

Steven Daugherty

(Reprinted From WindowsItPro Magazine)

How to Diagnose the Problem

A kernel can monopolize the CPU for several reasons, so the next step is to diagnose the problem to determine whether DPCs are in fact causing the high CPU usage and if so, which device is using those DPCs. You can use three tools to drill down and get the details needed for diagnosis:

- Performance Monitor (perfmon.exe). This is a built-in Windows tool
- Process Explorer (procexp.exe)
- Kernel Profiling Tool (kernrate.exe). This tool is part of the Microsoft Windows Server 2003 Resource Kit

To determine whether DPCs are causing high CPU usage, you can use Performance Monitor or Process Explorer. I find Process Explorer easier to use than Performance Monitor for this purpose, but I'll cover both in case you prefer to use Performance Monitor.

In Process Explorer, you can quickly and easily see whether DPCs are causing high CPU usage. In the main window, the System Idle Process is broken into three groups: Interrupts, DPCs, and System. Just doubleclick DPCs to bring up the DPCs properties page and select the Performance Graph tab.

The top two graphs show the DPCs' CPU current usage and CPU usage history, as Figure 2, page 53, shows.

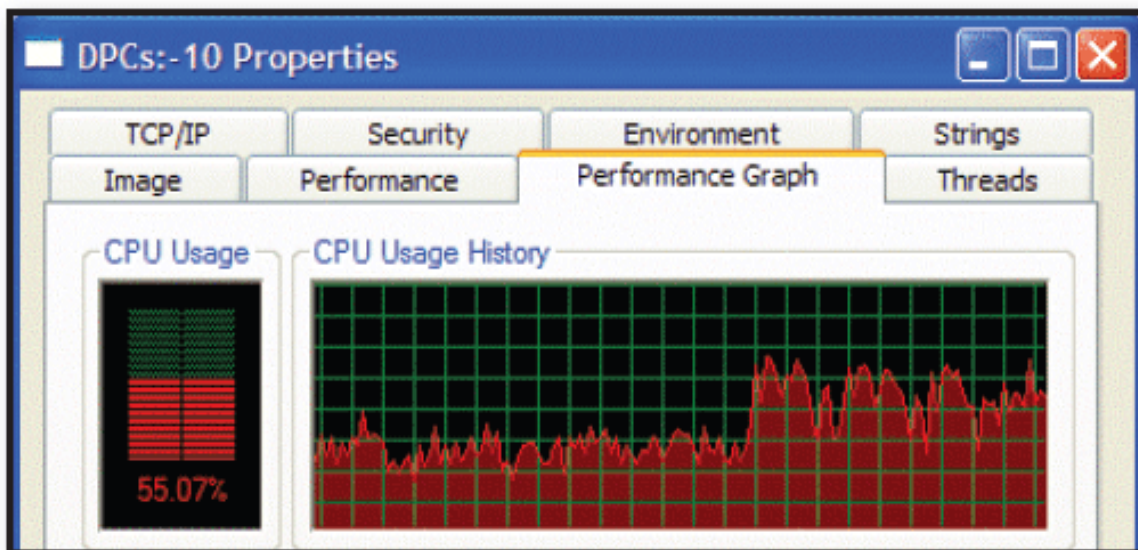


Figure 2: Using Process Explorer to determine how much time the processor is spending on DPCs

3 Steps to Troubleshooting Device Drivers

Steven Daugherty

(Reprinted From WindowsItPro Magazine)

Under the Processor object, Performance Monitor includes a counter named % DPC Time, which tells you the percentage of time a processor is spending on receiving and servicing DPCs. Comparing this percentage to the % Processor Time value reveals what portion of total CPU usage is consumed by a thread in the DPC queue, as Figure 3, page 53, shows. For information about how to use Performance Monitor, see "Performance Management in Windows" (March 2003, InstantDoc ID 37933).



Figure 3: Using Performance Monitor to determine how much time the processor is spending on DPCs

After Process Explorer or Performance Monitor has confirmed that the CPU is being disproportionately consumed by activity in the DPC queue, you can use the Kernel Profiling Tool to determine which driver is causing the problem. This command-line tool lists kernel modules, including kernel-mode device drivers and the percentage of kernel time that they're consuming.

The Kernel Profiling Tool has many arguments you can use, but for our purposes, you can run it without any arguments, with the command

kernrate

3 Steps to Troubleshooting Device Drivers

Steven Daugherty

(Reprinted From WindowsItPro Magazine)

After running this command, you first receive output like that in Figure 4. Wait for about 30 seconds or so, then press Ctrl+C. While the Kernel Profiling Tool runs, keep an eye on DPC activity in Process Explorer or Performance Monitor to ensure the CPU usage remains consistently high during the sample.

```
C:\>kernrate

Kernrate User-Specified Command Line:
kernrate

Kernel Profile (PID = 0): Source= Time,
Using Kernrate Default Rate of 25000 events/hit
Starting to collect profile data

***> Press ctrl-c to finish collecting profile data
```

Figure 4: Initial sample output from the Kernel Profiling Tool

Figure 5 shows an excerpt from the second part of the sample output. As you can see in the “Results for Kernel Mode” section, the problematic driver is intelppm, the Intel processor driver that’s part of the OS load. This driver throttles the CPU in order to conserve power consumption to extend battery life. The Intel processor driver probably wasn’t the cause of the problem; instead, the problem was likely due an interoperability issue with this driver and other hardware or drivers installed on my system. Regardless, it’s not required on a nonportable computer where battery life is of no concern.

3 Steps to Troubleshooting Device Drivers

Steven Daugherty

(Reprinted From WindowsItPro Magazine)

```
====> Finished Collecting Data, Starting to Process Results

-----Overall Summary:-----
PO      K 0:00:13.343 (34.6%)  U 0:00:01.812 ( 4.7%)  I 0:00:23.406 (60.7%)  DPC
0:00:09.234 (23.9%)  Interrupt 0:00:00.687 ( 1.8%)
Interrupts= 102538, Interrupt Rate= 2659/sec.

Total Profile Time = 38562 msec

Context Switches      ,      Total      Avg. Rate
System Calls         ,      161193,      4180/sec.
Page Faults          ,      1116268,      28947/sec.
I/O Read Operations  ,      37612,      975/sec.
I/O Write Operations ,      4498,      117/sec.
I/O Other Operations ,      277,      7/sec.
I/O Read Bytes       ,      13362,      347/sec.
I/O Write Bytes      ,      2025716,      450/ I/O
I/O Other Bytes      ,      1571044,      5672/ I/O
I/O Other Bytes      ,      1288736,      96/ I/O

-----

Results for Kernel Mode:
-----

OutputResults: KernelModuleCount = 170
Percentage in the following table is based on the Total Hits for the Kernel

Time 14302 hits, 25000 events per hit -----
Module Hits msec %Total Events/Sec
intelppm 8757 38567 61 % 5676485
ntoskrnl 4036 38567 28 % 2616226
win32k 459 38567 3 % 297534
hal 421 38567 2 % 272901
nv4_mini 372 38567 2 % 241138
USBPORT 103 38567 0 % 66766
nv4_disp 40 38567 0 % 25928
Ntfs 20 38567 0 % 12964
viexca2k 11 38567 0 % 7130
usbhcci 11 38567 0 % 7130
SYMEVENT 9 38567 0 % 5834
```

Figure 5: Identifying the problematic driver

How to Solve the Problem

The options for solving a DPC problem are to reinstall, update, or disable the driver that's causing the problem. Which option you choose depends on the driver. For example, you can't disable a driver that your system needs or update a driver when no updates exist. In my case, because the Intel processor driver isn't a required driver for my system, I decided to disable it.

There are several ways you can disable drivers, including through the registry, Device Manager, and Recovery Console (RC). I opted to disable the Intel processor driver through the registry. If you're interested in using RC or Device Manager to disable a driver, see the articles referenced in the "Learning Path."

3 Steps to Troubleshooting Device Drivers

Steven Daugherty

(Reprinted From WindowsItPro Magazine)

The registry data for the Intel processor driver is in the HKEY_LOCAL_MACHINE\SYSTEM\CurrentControl Set\Services\intelppm subkey. To disable this driver, I changed the Start entry from a value of 1 to a value of 4. Table 1 shows the Start values you can use for subkeys under the HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet Services key. For more information about how to read and change entries in these subkeys, see the Microsoft article “CurrentControlSet\Services Subkey Entries” (support.microsoft.com/kb/103000).

Table 1: Possible Start Values

Start Value	Description
0 (0x0)	Drivers required to read the boot volume are loaded by the kernel (Boot Loader).
1 (0x1)	Drivers are loaded at kernel initialization by the I/O subsystem.
2 (0x2)	Services are automatically started and drivers are automatically loaded at system startup by the Service Control Manager.
3 (0x3)	Services and drivers are available. The Service Control Manager starts the service or loads the driver only when initiated by user.
4 (0x4)	Services and drivers are disabled and can't be started or loaded.

As Microsoft continually reminds us, it's important that you understand how to restore the registry before you edit it. It's also important to reboot before and after editing the registry. Rebooting before a change is important to ensure a clean Last Known Good Configuration in case you need to restore the registry. (The Last Known Good Configuration includes everything under CurrentControlSet key and is updated after a successful logon.) Rebooting after a change is necessary for the change to take effect.

3 Simple Steps Is All It Takes

If your applications are loading slowly or don't load at all, the culprit might be a device driver and the DPCs it makes. By using tools such as Windows Task Manager, Performance Monitor, Process Explorer, the Kernel Profiling Tool, and the registry, you can quickly recognize, diagnose, and solve driver performance problems.