

# Powerful Tools for Diagnosing and Resolving Problems

BY RICK BYRLEY

**W**ith computers, and PCs in particular, the question is not if there is going to be a problem but when. Most of us spend a lot of time and effort preparing for the inevitable disk crash or system failure. However, sooner or later the best laid plans of mice and men go astray and we are left scratching our head trying to figure out what is wrong. Fortunately, OS/2 provides some powerful tools for diagnosing and resolving problems. This month's column highlights some of these tools.

## PSTAT

The PSTAT utility displays currently running processes, threads, system semaphores and active dynamic link libraries. PSTAT shows what is really going on behind the scene on your OS/2 system. As you've probably figured out by now, most of the programs running on your system do not show up in the window list. For example, most OS/2 machines are connected to some kind of network, but none of the network processes show up in the window list. PSTAT allows you to identify all processes currently running on a system, what their state is, and what runtime link libraries they are using.

So how can you use this information? Occasionally, you'll be able to resolve a DLL problem by cross-referencing what processes are using a particular DLL, or, more frequently, discovering the DLL being loaded is from a different location than you expect, such as from the current directory rather than from the LIBPATH statement in the CONFIG.SYS. PSTAT is also very useful for making sure a user hasn't misled you into assuming necessary software is running on the user's system when it in fact isn't. Have the user print out a PSTAT report just to make certain. And, finally,

you can use PSTAT to determine if you actually need to apply specific system DLL fixes posted by IBM.

Unfortunately, the PSTAT output to an OS/2 window is virtually unusable since lines wrap. It is better to redirect the output to a file (e.g., PSTAT > C:\PSTAT.TXT) and use an editor to display the information. You can also use the command switches /C, /S, /L, and /M to limit the display to information on current processes, system semaphores and link libraries or memory, respectively. You can also display more detailed information on a specific process by using the /P switch followed by the process id, e.g., PSTAT /P:003C.

---

**Most of us spend a lot of time and effort preparing for the inevitable disk crash or system failure. However, sooner or later the best laid plans of mice and men go astray and we are left scratching our head trying to figure out what is wrong.**

---

## RMVIEW

While PSTAT can assist you in resolving software problems on your system, it cannot help with the problems many of us dread the most: hardware configuration issues. While plug-and-play technology has helped, IRQ, DMA, I/O port and other device conflicts still seem to crop up frequently. Version 4.0 of OS/2 includes a nifty facility in the Hardware Setup folder

called the "Hardware Manager" that will display the currently installed devices with their respective IRQ, DMA, I/O and memory settings. Each device has an associated properties notebook that allows you to alter the device settings; simply double click on an object to display the notebook.

Although the "Hardware Manager" is convenient, it isn't present in any of the previous versions of OS/2. Even worse, it is a graphical program and therefore isn't available when you need it most — when your system is down and you are booted to floppies. In these situations you will want to use the text-based program RMVIEW.

The secret to using RMVIEW is learning to use the command switches that can be displayed by entering RMVIEW /?. Basically there are four views that display resource usage from the perspective of the physical device: the device driver, the logical device, or, in the world of plug-and-play, the detected device. You can limit the resources displayed to one particular resource using additional switches, including /IRQ for IRQ assignments, /IO for claimed Input/Output ports, and /DMA for DMA channel usage. For example, to display all IRQ assignments from the perspective of the physical devices, simply enter the command RMVIEW /P /IRQ.

Like PSTAT, the information from RMVIEW is difficult to use from an OS/2 window display. It's best to redirect the output to a file and use an editor to display the information.

## SYSLEVEL

The simplest of all the diagnostic commands is SYSLEVEL. This will display all system products currently installed along with the version and revision level. This information is useful in determining if

corrective service from IBM is available to resolve your problem.

### **SYSLOG**

If a problem is particularly nasty you may need to enable system logging. Although you typically will not be doing this without instructions from IBM support, there is no particular reason why anyone can't use the system logging facilities (other than they noticeably impact performance). To enable the system logging facility you must enter two statements in your CONFIG.SYS file. Assuming C: is the boot drive, enter the following

```
DEVICE=C:\OS2\LOG.SYS  
RUN=C:\OS2\SYSTEM\LOGDAEM.EXE
```

Then reboot your system. Attempt to recreate the problem, then open the logfile by issuing the command SYSLOG. This will produce a PM application that will display the system error log. Double-clicking on any entry will display details on the error, including possible causes and recommended actions, if available. This log is actually fairly impressive, although it is

sometimes disappointing when errors are not logged because the applications were not written to use the system log effectively. Applications must be written to use the system log, and unfortunately few use it effectively. Nevertheless, on a stubborn problem the system logging facility may yield that one piece of information that causes everything to fall into place.

### **UP AND RUNNING**

Keeping an OS/2 system up and running is usually a simple matter; like the Energizer bunny, OS/2 boxes just keep going and going and going. However, it never hurts to have a little insurance by learning to use the tools available if a problem should arise. **ts**

---

**Rick Byrley is senior workstation division technician for Softouch Systems, Oklahoma City, Okla., which provides both mainframe and PC software solutions. His primary focus is object-oriented programming. He can be reached at rbyrley@softouch.com.**

*©1998 Technical Enterprises, Inc. For reprints of this document contact sales@nasp.net.*

