

The Blue Screen of Death

Mark T. Edmead

(Reprinted from WindowsItPro Magazine)

You're working along just fine, and suddenly, your screen display changes from your nice user interface to something that looks like Screen 1. You know what it is: a Windows NT kernel STOP error, or the blue screen of death. So, what can you do? Often, the problem goes away when you reboot the system. But what if it doesn't? What does that screen mean? Is it safe to continue using the system? Let's look at what a kernel STOP error means, what can cause it, and most important, what information you can get from the blue screen.

```
*** STOP: 0x0000000a (0x00000000, 0x0000001a, 0x00000000, 0x00000000)
IRQL_NOT_LESS_OR_EQUAL

p4-0300 irq1:1f SYSVER: 0xf000030e

Dll Base DateStmp - Name Dll Base DateStmp - Name
80100000 2e53fe55 - ntoskrnl.exe 80400000 2e53eba6 - hal.dll
80010000 2e41884b - Rha154x.sys 80013000 2e4bc29a - SCSIPTOR.SYS
8001b000 2e4e7b6b - Scsidisk.sys 80220000 2e53f238 - Ntfs.sys
fe420000 2e406607 - Floppy.SYS fe430000 2e406618 - Scsiocdm.SYS
fe440000 2e406659 - Es Rec.SYS fe450000 2e40660f - Null.SYS
fe460000 2e4065f4 - Beep.SYS fe470000 2e406634 - Sermouse.SYS
fe480000 2e42a4a4 - i8042prt.SYS fe490000 2e40660d - Mouclass.SYS
fe4a0000 2e40660c - Kbdclass.SYS fe4c0000 2e40665e2 - VIDEOPRT.SYS
fe4b0000 2e53d49d - ati.SYS fe4d0000 2e40665e8 - vga.sys
fe4e0000 2e406655 - Msfs.SYS fe4f0000 2e414f30 - Npfs.SYS
fe510000 2e53f222 - NDIS.SYS fe500000 2e40719b - elnkii.sys
fe550000 2e406697 - TDI.SYS fe530000 2e47c740 - nbf.sys
fe560000 2e5279d9 - nwlknkpx.sys fe570000 2e53a89e - nwlknkb.sys
fe580000 2e494973 - tcPIP.sys fe5a0000 2e5256b8 - afd.sys
fe5b0000 2e5279d3 - netbt.sys fe5d0000 2e4167f7 - netbios.sys
fe5e0000 2e4066b3 - mup.sys fe5f0000 2e4f9f51 - rdr.sys
fe630000 2e53f24a - srv.sys fe660000 2ef16062 - nwlknkpx.sys

Address dword dump Build [1057] - Name
FF541E4c fe5105df fe5105df 00000001 ff640128 fe4a8228 000002fe - NDIS.SYS
ff541e60 fe501368 fe501368 00000246 00004002 00000000 00000000 - elnkii.sys
ff541eb4 fe481509 fe481509 ff6688c8 ff668288 00000000 ff668138 - i8042prt.SYS
ff541ee0 fe481ea8 fe481ea8 fe482078 00000000 ff541f04 8013c58a - i8042prt.SYS
ff541ee4 fe482078 fe482078 00000000 ff541f04 8013c58a ff6688c8 - i8042prt.sys
ff541ef0 8013c58a 8013c58a ff6688c8 ff668040 80405900 00000031 - ntoskrnl.exe
ff541efc 80405900 80405900 00000031 06060606 06060606 06060606 - hal.dll

Restart and set the recovery options in the system control panel
or the /CRASHDEBUG system start option if this message reappears,
contact your system administrator or technical support group.
CRASHDUMP: Initializing miniport driver
CRASHDUMP: Dumping physical memory to disk: 2000
CRASHDUMP: Physical memory dump complete
```

What Happens at the Kernel Level?

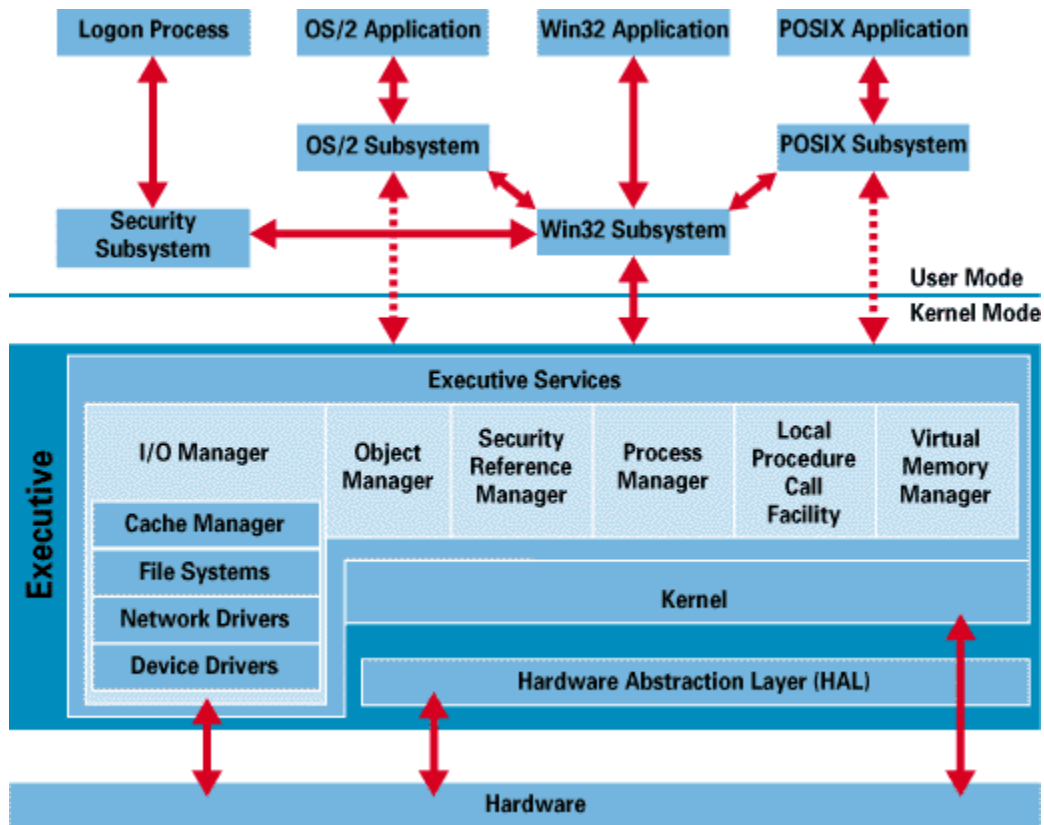
First, let's review the basics of the NT architecture. The NT operating system has two layers: user mode and kernel mode. User mode is where the various subsystems--such as the Win32, POSIX, or OS/2 subsystem--reside. Components in this mode provide the environments in which all user applications run. For instance, Win32 programs run on the Win32 subsystem.

As you see in Figure 1, the kernel mode sits between the user mode and the physical layer (the hardware) and prevents the user mode from directly accessing the hardware. The kernel mode also is the home for the various NT executive services, such as the Object Manager, Security Reference Monitor, and Process Manager. Just above the physical device hardware lies the hardware abstraction layer (HAL) and above that is the NT microkernel. The HAL is the portion of the kernel that is written in the specific platform assembly language. The microkernel is the heart of the OS that takes care of all the NT internal OS operations.

The Blue Screen of Death

Mark T. Edmead

(Reprinted from WindowsItPro Magazine)



An important component of executive services is the I/O Manager. Besides taking care of all input and output for the operating system, the I/O Manager manages communications between drivers and supports all file system drivers and hardware device drivers.

NT is a modular operating system; this fact means you can add DLLs or device drivers to add capabilities to the system. You can, for instance, add fault tolerance to NT by adding device drivers. When a peripheral manufacturer develops a driver for NT, the driver is most likely a kernel mode driver: It resides in the kernel mode area and probably interfaces with Microsoft kernel drivers. You can think of kernel drivers as the NT counterpart to Windows 3.1 or NT virtual device drivers (VxDs). Kernel drivers are the low-level mechanisms for talking to the hardware. So when the driver does something it's not supposed to, the error occurs at the lowest level and directly affects the overall system and causes a kernel STOP error.

If an application operating in user mode does something to cause an error, NT halts the process and generates an Illegal Operation error. Because every Win32 application has its own virtual protected space, this error condition doesn't affect any other Win32 programs running. If the application tries to directly access the hardware without going through the correct methods, NT notices this and generates an exception error. A nice thing about NT is that it has good protection systems for erratic applications.

When an application faults, you can close the offending program and resume work. Kernel error conditions, however, typically are not recoverable; you have to reboot the system. You can think of the kernel STOP as a built-in error-trapping mechanism. A kernel STOP error is NT's way of halting further activity before the activity severely damages your system or corrupts data.

The Blue Screen of Death

Mark T. Edmead

(Reprinted from WindowsItPro Magazine)

What Does This Weird Screen Mean?

OK, so what does this screen tell me? The kernel STOP may mean that a kernel driver--either a system device driver or a third party driver--has illegally accessed the privileged kernel area. Or the kernel STOP may mean that you have mixed SIMMs or added a bad network controller or SCSI controller. In these cases, you can fix the problem by removing the offending hardware device. If you have not added any new hardware, you need to get more information from the blue screen. Let's look at each portion of Screen 1. Fortunately, you don't need to understand everything on the screen.

At the top of the display is a hexadecimal value followed by four hex numbers in parentheses. The first hex code is the kernel error code. With this error code, you can determine where the error occurred, but not which driver caused the error. Table 1 lists the various error conditions. In our example, the error condition is 0*0000000A, IRQL_NOT_LESS_OR_EQUAL. This code means that a process attempted to access pageable memory at a process internal request level (IRQL) that was too high. Microsoft Windows NT Server Resource Kit and Microsoft Windows NT Workstation Resource Kit have complete listings of STOP codes.

TABLE 1: Kernel Mode Error Conditions

0X0000000A

IRQL_NOT_LESS_OR_EQUAL

A process attempted to access pageable memory at a process internal request level (IRQL) that was too high. A process can access only objects that have priorities (IRQL) equal to or lower than its own. A device driver using improper addresses usually is the cause of this error.

0X00000019

BAD_POOL_HEADER

This error, also called Invalid Pool Header, can appear for several reasons. You can find the cause by debugging the system.

0X0000001E

KMODE_EXCEPTION_NOT_HANDLED

This error is a very common bug check. Usually the exception address (the second parameter) pinpoints the driver or function that caused the problem. Always note this address and the link date of the driver or image that contains this address.

0X00000024

NTFS_FILE_SYSTEM

All file system bug checks have encoded the source file and the line within the source file that generated the bug check in their first ULONG (unsigned long value). The upper 16 bits identify the file; the lower 16 bits identify the source line in the file where the bug check occurred.

0X00000051

REGISTRY_ERROR

Something has gone terribly wrong with the Registry. It might have received an I/O error while attempting to read one of its files as a result of a hardware problem or file system corruption.

0X00000077

KERNEL_STACK_INPAGE_ERROR

The system could not read in the requested page of kernel data. A bad block in a paging file or a disk controller error might be the cause. If the error is a result of a paging error, AUTOCHK will attempt to

The Blue Screen of Death

Mark T. Edmead

(Reprinted from WindowsItPro Magazine)

map out the bad block when you restart the system. The second parameter identifies the cause of the error:	
0XC000009A	
signifies a lack of nonpaged pool resources.	
0XC000009C and 0XC000016A	
both signify a bad block on the drive.	
0XC0000185	
signifies improper termination of a SCSI device, bad SCSI cabling, or two devices attempting to use the same IRQ.	
0X0000007A	KERNEL_DATA_INPAGE_ERROR
See 00000077, KERNEL_STACK_INPAGE_ERROR	
0X0000007B	INACCESSIBLE_BOOT_DEVICE
The system cannot access the boot device. Often, this message signifies a disk controller configuration problem or an error in accessing the hard disk. Another possible cause is that, during initialization of the I/O system, the driver for the boot device failed to initialize the boot device (device not available, SCSI error). Another possibility is that the file system could not recognize the data on the boot device. The message also might mean that a virus has infected the boot sector.	
0X0000007F	UNEXPECTED_KERNEL_MODE_TRAP
A trap that the kernel doesn't have permission to have or catch occurred in privileged processor (kernel) mode. The message may signify a computer RAM problem (mismatched SIMMs), a BIOS problem, or corrupted file system drivers. The first number in the bug check is the number of the trap. Consult an Intel x86 Family manual for the trap codes.	
0X00000080	NMI_HARDWARE_FAILURE
A hardware error has occurred, in which HAL reports the information that it can identify and directs the user to call the hardware vendor.	

If the stop code identifies a problem such as a 0X0000007B INACCESSIBLE_BOOT_DEVICE error—that you can resolve locally, correct the problem (e.g., remember to turn on the external hard disk drive) and then restart the computer to verify the resolution. If you can't resolve the error easily, you might be able to use a kernel debugger to isolate the problem.

The values in the parentheses give more specific information about what the driver was doing when the error happened. The first value (00000000) points to the address that the driver referenced improperly. The second value is the IRQL that was required to access the memory. The third value specifies whether the driver was doing a read or a write. The fourth value points to the instruction address that attempted the access. By looking at the STOP code and the third and fourth parameters, you can possibly determine what caused the error condition.

The Blue Screen of Death

Mark T. Edmead

(Reprinted from WindowsItPro Magazine)

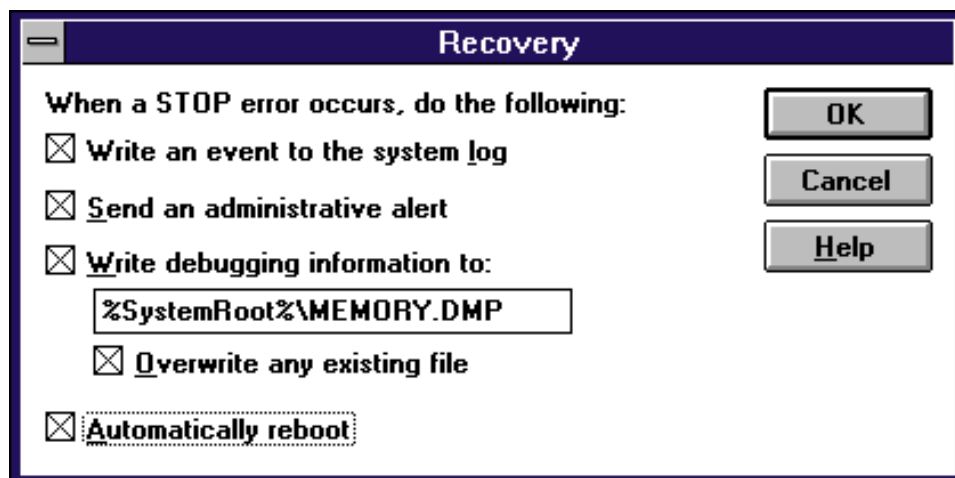
The information in the middle of the screen, called the DLL base (starting with 80100000), lists the drivers the system loaded and initialized successfully. The bottom of the screen, called the DLL load base, shows the drivers in the stack. The first driver in the list is the next one to be pushed from the stack, or executed. In many cases, the first driver is the offending driver. When the base address of the first driver is close to the fourth value at the top of the screen (the instruction address that attempted access), you can hypothesize that the driver might have caused the problem when it was initializing and being pushed off the stack. In Screen 1, the number in the DLL load base (000002fe) is very near to the fourth value (00000000) at the top of the screen.

Not all blue screens are easy to read. In this example, the problem driver might still be a driver listed in the middle of the screen, even though the screen shows that it initialized correctly, or the driver might not be on the screen at all (in the case of a bad controller card). Or something other than a driver might have caused the problem. When you can't easily find the problem, you need to go to the next step: debugging.

How to Debug

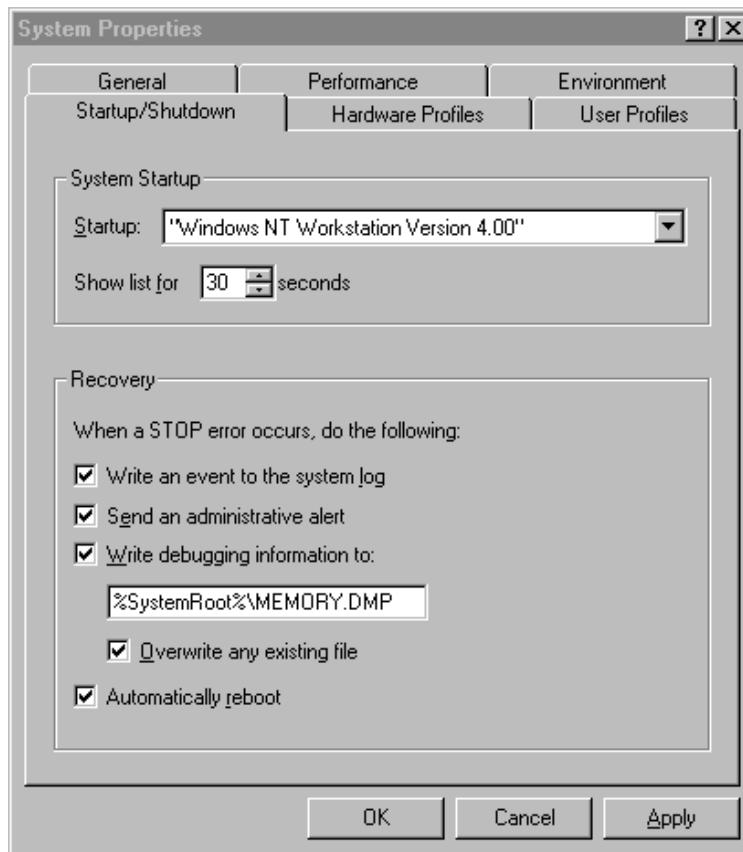
Let's assume that you have determined that the cause of the kernel STOP is an installed device driver and not a hardware problem. Now what? Well, it depends. If you are not the developer of the driver, you probably want to save the NT image information, and let someone else figure out what happened. This approach is called noninteractive debugging. If you are the developer and you have the source code, you can use the kernel debugger that NT provides to step through the driver code. This technique is interactive debugging.

Noninteractive debugging. NT gives you the option of saving the image of the operating system (at the time of the kernel STOP error) to your hard disk. You can use this information to determine the cause of the problem. To save the NT image to disk, go to the Control Panel, System applet (Screen 2, shows the NT 3.51 setup, and Screen 3, the NT 4.0 setup). You need to be an Administrator to access the options. You can write the event to the system log so you can view the error in the Event Viewer. This option is handy, because if you set your system to reboot automatically after a kernel STOP error, the condition may go unnoticed. You also have the option to have the system send an administrative alert. This alert is useful, for example, when the server has the kernel STOP error and you are working someplace where you can't see the server screen.



The Blue Screen of Death

Mark T. Edmead
(Reprinted from WindowsItPro Magazine)



The next option lets you write the memory dump file to %SystemRoot%\MEMORY.DMP. Note that the size of the image file is roughly the size of your physical RAM. Therefore, if you have 128MB of RAM, your dump file will be 128MB! You can select the option to overwrite the existing file, if one already exists. The last option is to set the system to automatically reboot. If you elect to save the image file and to reboot, the process may take a while, depending on RAM size. I have seen this process take more than 20 minutes, so be patient.

You might also want to have the computer send an administrative alert. An alert is useful when the system that has the problem is not near you, and you need to be informed when the error occurs. You can configure administrative alerts in the Control Panel, Server applet.

The Windows NT Server and NT Workstation CD-ROMs contain some tools to help you with this memory file. dumpflop.exe writes the memory file to floppies (a 32MB memory file fits on about 10 disks). Unfortunately, Microsoft does not accept the memory file on any other medium. Once you have created the dump file, you can make it available to a Microsoft Product Support Specialist either by sending the floppies to Microsoft or by preparing a Remote Access Service (RAS) connection for Microsoft Product Support to dial in and view the file contents remotely. Or you can submit the file to Microsoft over the Internet by connecting to ftp.microsoft.com and copying the file to /transfer/incoming/bussys/winnt.

You can use another utility, dumpchk.exe, to examine the integrity of the dump file and verify that the system created the file correctly. With dumpchk, you can view basic information about the dump file, such as which NT version was running and the STOP error codes.

The Blue Screen of Death

Mark T. Edmead

(Reprinted from WindowsItPro Magazine)

Another useful utility is dumpexam.exe, which converts the memory file into a readable text file. You need three files to run dumpexam: dumpexam.exe, imagehlp.dll, and for the Intel platform, kdextx86.dll (the third file depends on the platform). The three files must be in the same directory. You can find them on the CD-ROM of the NT Server or the NT Workstation CD-ROM in the directory \support\debug\<platform>, where platform is i386, alpha, mips, or ppc.

The noninteractive debugging method is ideal for users who don't want to debug the driver, but just want to figure out which one is at fault. To run dumpexam, you need to load the symbol files, which contain NT system debugging information. Make sure that the symbol files are for the version of NT you're running, including any installed service packs. For the Intel version of NT, the symbol files are in the \support\debug\i386\symbols directory on the NT resource kits' CD-ROMs

Figure 2 shows the syntax for dumpexam. For example, if you want to analyze a dump file for a computer with NT Workstation 4.0, the symbols are in the directory d:\symbols. The dump file, server.dmp, is in the directory d:\dump. The command line reads

```
dumpexam -y d:\symbols d:\dump\server.dmp
```

The results of the exam will be in %SystemRoot%\MEMORY.TXT.

FIGURE 2:

Syntax for dumpexam

```
dumpexam [options] [CrashDumpFile]
```

where

-? displays the command syntax

-v specifies verbose mode

-p prints the header only

-f filename specifies the output file nameX

-y path sets the symbol search path

XYou need to specify the dump file path with this option only if you have moved the dump file.

Interactive debugging. The other method of debugging is interactive debugging. Device driver developers, rather than systems administrators, usually prefer interactive debugging because the process requires extensive knowledge of NT internals.

Interactive debugging requires you to have another PC (a host machine) with NT installed and to run the kernel debugger on the host machine. It must be running the same version of NT as the target machine. The host machine must be connected to the problem computer via a modem or null cable connection.

Is the System Safe?

Can you safely use the system after a kernel STOP error? The answer depends on whether you can isolate what caused the problem. I've seen cases where the error condition happens once, never to repeat again. In other cases, however, the error occurs after the user has installed or updated a driver. In this case, you need to remove the driver and start over. When you get the kernel STOP error, reboot the system, and hit the space bar when you see the Last Known Good text. This action starts NT with the last known working configuration, without the offending driver. The option of reverting to the last known working configuration reinforces the wisdom of installing one driver at a time and

The Blue Screen of Death

Mark T. Edmead

(Reprinted from WindowsItPro Magazine)

making sure that the driver works before you install another driver. If the driver doesn't work correctly, you can revert to the previous working configuration. If you install two or more drivers at the same time and one of them causes a problem, you will have trouble determining which driver caused the problem.

If the problem is not related to a driver, look at new system hardware, such as a new controller card. To determine whether a controller card is the problem, remove the card and test the system again. If the problem goes away, check whether the card (or any new hardware you add to your system) is in the Microsoft Hardware Compatibility List (HCL). TechNet and Microsoft's Web site (<http://www.microsoft.com>) have an up-to-date list.

The Bottom Line

I hope that, after reading this article, the blue screen won't intimidate you. Using the techniques I've explained, you can find out, in general, why you got the screen and perhaps tell specifically what caused the problem. If you aren't a device driver developer and don't want to deal with interactive system debugging, noninteractive is the way to go. Your goal is to get your system up and running as fast as possible. Isolating the problem is the first step. For additional resources, see the sidebar, "Other Sources of Help."