Workstation

CHAPTER

Running Applications on Windows NT

21

About Chapter 21

This chapter explores the various application environments supported by Windows NT 4.0, as well as some techniques for customizing application performance.

Chapter 21 begins by explaining that Windows NT 4.0 is designed to run applications created for many different operating system environments, including MS-DOS, Win16, Win32, POSIX, and OS/2. Next, the chapter presents the various hardware platforms on which Windows NT 4.0 supports applications.

Following that is an exploration of application priorities, and how to use the Start command to start applications at various priorities. Configuring foreground application priority boost is also explained.

Next, the chapter details how to use Windows NT Task Manager to perform various tasks, such as stopping an application, viewing performance statistics, and changing an application's base priority.

Finally, Chapter 21 presents tips on troubleshooting common application problems.

This chapter includes a review activity that will give you practice troubleshooting application problems. It also includes a hands-on lab during which you'll configure foreground application responsiveness and start applications at various priorities on your Windows NT Workstation computer.

This chapter contains some valuable and interesting information, but is optional reading unless you're preparing for the Workstation exam. The information in this chapter covers both objectives in the Running Applications section in the Workstation exam's objectives.

Application Environments

Windows NT 4.0 is designed to run applications created for several different types of operating system environments. Windows NT supports these different application types by using multiple environment *subsystems*. These subsystems each include the *application programming interface* (API) of the operating system or environment that the subsystem is designed to support. The subsystems enable applications to run in the Windows NT environment as if they were running in the operating system environment they were designed for.

The application types and operating system environments supported by Windows NT 4.0 include:

- MS-DOS applications (MS-DOS environment)
- 16-bit Windows applications, such as those written for Windows 3.*x* and Windows for Workgroups (Win16 environment)
- 32-bit Windows applications, such as those written for Windows NT and Windows 95 (Win32 environment)
- POSIX applications (POSIX environment)
- OS/2 applications, such as those written for OS/2 1.*x* (OS/2 environment)

Each of these environments is discussed in the following sections.

MS-DOS Environment

Applications designed for the MS-DOS environment are typically legacy applications that use a character-based, command-line interface. A *character-based, command-line interface* is one that relies on keyboard input rather than mouse input. Additionally, the screen display does *not* necessarily match the printed output—it's not *What You See Is What You Get* (WYSIWYG). Many utilities designed for MS-DOS are still useful even though they haven't been rewritten for use in the Windows graphical environment.

Windows NT 4.0 includes support for MS-DOS applications via a subsystem called an *NT Virtual DOS Machine* (NTVDM). An NTVDM emulates an Intel 486 computer running the MS-DOS operating system. NTVDM support is included for all hardware platforms supported by Windows NT 4.0, including Intel 486 and higher, DEC Alpha, PowerPC, and MIPS R4000.

Most MS-DOS applications are supported by Windows NT in an NTVDM. However, MS-DOS applications that make direct calls to hardware are *not* supported by Windows NT. These applications will not run correctly in a Windows NT environment.

Each MS-DOS application runs in its own separate NTVDM. Because each application runs in its own separate NTVDM, if an MS-DOS application crashes, other applications are *not* affected.

Windows NT 4.0 enables multiple NTVDMs to be run. Because each MS-DOS application runs in a separate NTVDM, Windows NT can preemptively multitask multiple MS-DOS applications. In *preemptive multitasking*, the operating system allocates processor time between applications. Because Windows NT, *not* the application, allocates processor time between multiple applications, one application can be preempted by the operating system, and another application allowed to run. When multiple applications are alternately paused and then allocated processor time, they appear to run simultaneously to the user.

NTVDMs have three threads. (A *thread* is the smallest unit of processing that can be scheduled by the Windows NT Schedule service. All applications require at least one thread.) Two of these threads are used to maintain the NTVDM environment. The third thread is used by the application. An application that runs in an NTVDM is referred to as a singled-threaded application (because only *one* thread is used by the application).

Some MS-DOS applications require environmental settings that would normally be configured in the MS-DOS computer's `Autoexec.bat` or `Config.sys` files. For example, a path to the application may need to be specified, or a *terminate-and-stay-resident (TSR)* program may need to be loaded prior to starting the application. To provide the same environmental settings in a Windows NT environment, you can edit the `Autoexec.nt` and/or `Config.nt` files to include any necessary instructions. Settings contained in the `Autoexec.nt` and `Config.nt` files are executed each time an NTVDM is started. These files are edited in the same manner as you would edit an `Autoexec.bat` or `Config.sys` file. The `Autoexec.nt` and `Config.nt` files are stored in the *<winntroot>*`\System32` folder.

Win16 Environment

Win16 environment applications consist of 16-bit Windows applications designed for Windows 3.*x* and Windows for Workgroups. These applications are graphical applications that accept input from both a mouse and keyboard. Often the screen display matches the printed output (WYSIWYG).

Windows NT provides support for 16-bit Windows applications via a special subsystem called *WOW*, for *Win16-on-Win32*. The WOW subsystem emulates an Intel 486 computer running MS-DOS and Windows 3.1. The WOW subsystem runs in an NTVDM called the *Win16 NTVDM*. Because NTVDMs are supported on all Windows NT platforms, Windows NT supports Win16 applications on all hardware platforms supported by Windows NT.

Most 16-bit Windows applications are supported by Windows NT. However, 16-bit Windows applications that make undocumented calls to the operating system or that require specific device drivers that make direct calls to hardware may not run correctly on Windows NT.

By default, when multiple Win16 applications are run at the same time, they all run in a single Win16 NTVDM. This means that, by default, all Win16 applications share the same memory space and the Win16 NTVDM's single thread. Because the Win16 applications share the same memory space, if one application crashes, other Win16 applications may also crash. Because multiple Win16 applications share a single NTVDM's thread, Windows NT *can't* preemptively multitask multiple Win16 applications.

To prevent a rogue Win16 application from crashing all of your other Win16 applications, and to allow Win16 applications to be preemptively multitasked, Windows NT permits Win16 applications to be run in separate NTVDMs. This is referred to as *running Win16 applications in separate memory spaces.*

Running Win16 applications in separate memory spaces

Windows NT allows the user to run a 16-bit Windows application in its own separate memory space. When a Win16 application is configured to run in its own separate memory space, Windows NT assigns that application its own Win 16 NTVDM when the application is run.

Running Win16 applications in separate memory spaces has advantages and disadvantages. Advantages include:

- When a Win16 application running in a separate memory space crashes, other applications are *not* affected.
- Windows NT can preemptively multitask Win 16 applications when they are run in separate memory spaces.

Disadvantages include:

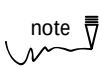
- More RAM and system resources are used when Win 16 applications are run in separate memory spaces.
- Some Win 16 applications that use shared memory instead of Object Linking and Embedding (OLE) or Dynamic Data Exchange (DDE) to communicate with other Win 16 applications may not work correctly when run in separate memory spaces.

There are two methods you can use to run a Win16 application in a separate memory space: You can configure the properties of the shortcut to the application, or you can use a batch file that uses the `Start/separate` command.

Configuring the properties of the shortcut to the Win16 application is the easiest and most commonly used method of configuring a Win16 application to run in a separate memory space.

TO CONFIGURE THE PROPERTIES OF THE SHORTCUT TO A WIN16 APPLICATION, FOLLOW THESE STEPS:

1. On the desktop or in Windows NT Explorer, right-click the shortcut to the Win16 application. Select Properties from the menu that appears.
2. The application's Properties dialog box appears. Click the Shortcut tab.
3. The Shortcut tab appears, as shown in Figure 21-1. Notice the Run in Separate Memory Space check box.



If the Run in Separate Memory Space check box is selected and grayed out, the application is a Win32 (not a Win16) application, and will always be run in a separate memory space.

If the Run in Separate Memory Space check box is available, select it. The application is now configured to run in a separate memory space. Click OK.

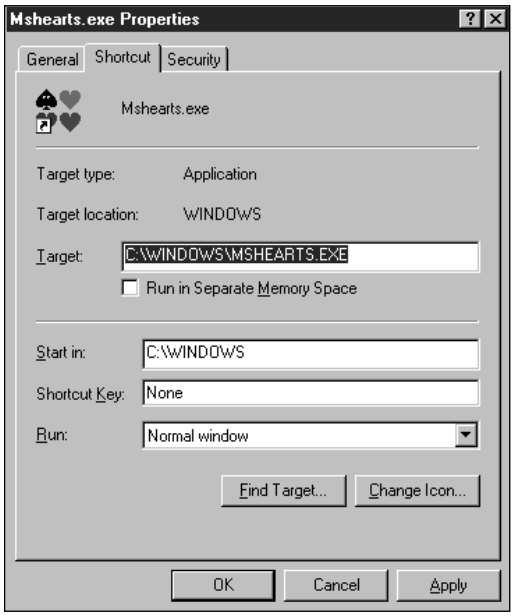


FIGURE 21-1 **Configuring a Win16 application to run in a separate memory space**

Win32 Environment

The Win32 environment is Windows NT's native application environment. It is the preferred and fastest environment for running applications on Windows NT 4.0, because no emulation or workarounds are required. Win32 environment applications consist of 32-bit Windows applications written specifically for Windows NT and/or Windows 95. Windows NT provides support for Win32 applications via the *Win32 subsystem*. Windows NT supports the Win32 subsystem on all hardware platforms supported by Windows NT.

Win32 applications are *source-compatible* across all supported hardware platforms. This means that Win32 applications must be recompiled for each hardware platform in order to be run on that platform.

Each Win32 application runs in its own separate memory space. Because of this, if a Win32 application crashes, other applications are *not* affected. Windows NT can preemptively multitask multiple Win32 applications.

POSIX Environment

Portable Operating System Interface for Computing Environments (POSIX) was developed as a set of accepted standards for writing applications for use on various UNIX computers. POSIX environment applications consist of applications developed to meet the POSIX standards. These applications are sometimes referred to as *POSIX-compliant applications*.

Windows NT provides support for POSIX-compliant applications via the *POSIX subsystem*. Windows NT supports the POSIX subsystem on all hardware platforms supported by Windows NT. To fully support POSIX-compliant applications, at least one NTFS partition is required on the Windows NT computer.

POSIX applications are source-compatible across all supported hardware platforms. This means that POSIX applications must be recompiled for each hardware platform in order to run on that platform.

Each POSIX application runs in its own separate memory space. Because of this, if a POSIX application crashes, other applications are *not* affected. Windows NT can preemptively multitask POSIX applications.

OS/2 Environment

OS/2 environment applications consist of 16-bit, character-based applications designed for OS/2 version 1.*x*. Applications designed for other versions of OS/2, including OS/2 2.*x*, 3.*x*, and Presentation Manager applications, are *not* supported by Windows NT. Microsoft has developed an add-on product that allows Windows NT to support OS/2 1.*x* Presentation Manager applications. This product is not included with Windows NT—it must be purchased separately from Microsoft.

Windows NT provides support for OS/2 applications via the *OS/2 subsystem*. It supports the OS/2 subsystem only on Intel 486 and higher platforms. Windows NT does *not* support the OS/2 subsystem on any other hardware platforms.

However, some OS/2 applications, called *real-mode applications*, can be run in an MS-DOS environment. Because Windows NT supports MS-DOS NTVDMs on *all* hardware platforms that it supports, real-mode OS/2 applications can be run in an NTVDM on any of these platforms by using the `Forcedos.exe` command to start the application.

Each OS/2 application runs in its own separate memory space. This means that if an OS/2 application crashes, other applications are *not* affected. Windows NT can preemptively multitask OS/2 applications.

Application Support on Diverse Hardware Platforms

Windows NT 4.0 supports several different types of applications on various hardware platforms. Not all application types are supported on every hardware platform.

Some types of applications are source-compatible across hardware platforms. As mentioned previously, this means that the application must be recompiled for each hardware platform that you want to run it on. For example, to run a POSIX 1.*x* compliant application on both the Intel and DEC Alpha platforms, the application must be compiled separately for each processor type.

Table 21-1 shows the support provided by Windows NT 4.0 for different application types on diverse hardware platforms. Each hardware platform is based on a different type of processor.

TABLE 21-1 APPLICATION SUPPORT ON WINDOWS NT 4.0 COMPUTERS WITH VARIOUS PROCESSORS

Application Type	*How the Application Runs in Windows NT (Broken Down by Processor Type)*			
	Intel 486 or Higher	*DEC Alpha*	*PowerPC*	*MIPS R4000*
MS-DOS and Win16	Runs in a Virtual DOS Machine (NTVDM)	Runs in Intel 486 emulation mode	Runs in Intel 486 emulation mode	Runs in Intel 486 emulation mode
Win32	Runs in Win32 subsystem; source code must be recompiled for Intel processor	Runs in Win32 subsystem; source code must be recompiled for DEC Alpha processor	Runs in Win32 subsystem; source code must be recompiled for PowerPC processor	Runs in Win32 subsystem; source code must be recompiled for MIPS R4000 processor

continued

TABLE 21-1 *(continued)*

Application Type	How the Application Runs in Windows NT (Broken Down by Processor Type)			
	Intel 486 or Higher	DEC Alpha	PowerPC	MIPS R4000
POSIX	Runs in POSIX subsystem; source code must be recompiled for Intel processor	Runs in POSIX subsystem; source code must be recompiled for DEC Alpha processor	Runs in POSIX subsystem; source code must be recompiled for PowerPC processor	Runs in POSIX subsystem; source code must be recompiled for MIPS R4000 processor
OS/2 1.*x* character-based	Runs in OS/2 subsystem; only OS/2 1.*x* character-based applications are supported—additional software is required to support other OS/2 applications	No OS/2 application support; however, real-mode OS/2 applications can be run in an MS-DOS NTVDM by using the `Forcedos.exe` command to start the application	No OS/2 application support; however, real-mode OS/2 applications can be run in an MS-DOS NTVDM by using the `Forcedos.exe` command to start the application	No OS/2 application support; however, real-mode OS/2 applications can be run in an MS-DOS NTVDM by using the `Forcedos.exe` command to start the application

Application Priorities

The Windows NT Schedule service uses application priorities to determine which applications receive the most processor time. Applications that have a high priority receive more processor time than applications with a low priority.

An *application priority* is a number between 0 and 31 that is assigned to an application when it is started. By default, most user applications are assigned a priority of 8, the normal priority. A user application can be assigned a priority between 0 and 15. These applications can be written to a paging file.

A real-time or kernel mode application can be assigned a priority between 16 and 31. Real-time applications *can't* be written to a paging file. When running, these applications are always stored in RAM.

Windows NT can dynamically raise and lower application priorities based on changing conditions in the computer. Most application priority changes are beyond the user's control, and are managed by the operating system. However, the base priority assigned to an application can be set by the user, and the foreground application can be assigned a one- or two-point boost in priority by the user.

Starting Applications at Various Priorities

In Windows NT, the `Start` command is used to start applications at various priorities. The `Start` command can be used in batch files and from the command prompt. The `Start` command *can't* be used in shortcuts to applications.

Six switches are commonly used with the Windows NT `Start` command. These switches are listed and described in Table 21-2.

TABLE 21-2 WINDOWS NT START COMMAND SWITCHES

SWITCH	*DESCRIPTION*
`/low`	Starts the application with a base priority of **4**.
`/normal`	Starts the application with a base priority of **8**. This is the priority that is normally assigned to most user applications. Windows NT typically starts user applications with a base priority of 8 when no other priority is specified.
`/high`	Starts the application with a base priority of **13**.
`/realtime`	Starts the application with a base priority of **24**. An application with this priority *can't* be written to a paging file. Applications started at the real-time base priority can slow the performance of the operating system itself. The real-time base priority should be used with extreme caution and is not recommended for most applications.
`/min`	Does *not* affect the base priority of an application. It starts an application in a minimized window. This switch can be used in conjunction with a priority switch and/or the `/separate` switch.
`/separate`	Does *not* affect the base priority of an application. It starts a Win16 application in a separate memory space.

For example, to start User Manager at a high priority and in a minimized window, the following command is used at the command prompt:

```
Start /min /high Usrmgr.exe
```

It's been my experience that the `Start` command often doesn't work properly unless it is run from the application's default folder. For example, to start User Manager, the `Start` command should be run from the *<winntroot>*`\System32` folder.

If you are unable to change an application's base priority by using the `Start` command, even from the application's default folder, you may need to use Windows NT Task Manager to change the application's base priority after it is started. (Task Manager is covered later in this chapter.)

Configuring Foreground Application Priority Boost

Normally it is desirable to give the application running in the foreground (the active application) a higher priority than applications running in the background. By default, Windows NT assigns a two-point priority boost to the foreground application.

You can boost the foreground application's priority by zero, one, or two points. The foreground application priority boost is *applied* to an application when it becomes the foreground application, and is *removed* from that application when the application is minimized or when another application becomes the foreground application. Once the foreground application priority boost is configured, that boost is applied to the foreground application from then on until the foreground application priority boost is changed.

The System application in Control Panel is used to configure the amount of boost a foreground application is assigned by Windows NT, as the next section explains.

TO CHANGE THE FOREGROUND APPLICATION'S PRIORITY BOOST, FOLLOW THESE STEPS:

1. Select Start ≫ Settings ≫ Control Panel.
2. In Control Panel, double-click the System icon.
3. The System Properties dialog box appears. Click the Performance tab.
4. The Performance tab appears, as shown in Figure 21-2. Notice that the boost slide bar is set at Maximum, by default.

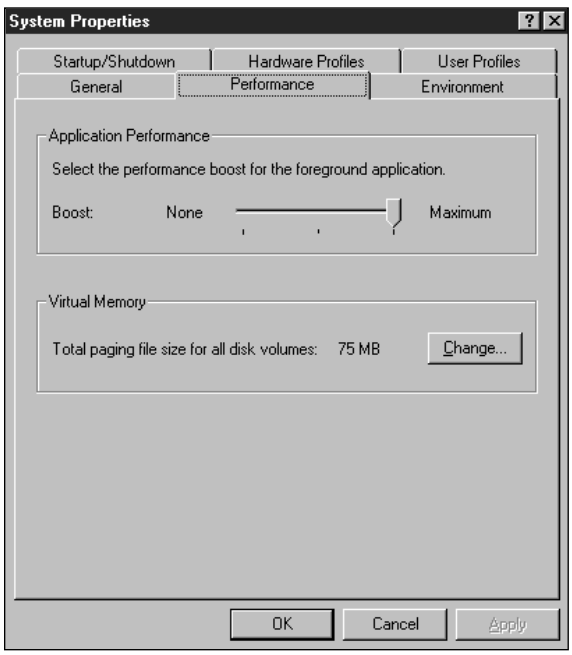


FIGURE 21-2 **Configuring foreground application performance boost**

Use the boost slide bar in the Application Performance section to choose one of three possible settings:

- **None**—No foreground application priority boost is applied.
- **Middle Setting**—A one-point application priority boost is applied to the foreground application.
- **Maximum**—A two-point application priority boost is applied to the foreground application. This is the default setting. Click OK.

5. The Control Panel dialog box reappears. Exit Control Panel.

Using Windows NT Task Manager

Windows NT Task Manager is a Windows NT administrative utility that can be used to start and stop applications; to view performance statistics, such as memory and CPU usage; and to change a process's base priority.

There are four different ways to access Task Manager:

- By pressing Ctrl + Shift + Esc
- By pressing Ctrl + Alt + Delete, and then clicking the Task Manager command button in the Windows NT Security dialog box;
- By right-clicking a blank space on the taskbar (on the Windows NT desktop), and then selecting Task Manager from the menu that appears;
- By selecting Start ➪ Run, and then typing **taskmgr** in the Run dialog box.

Using Task Manager to Stop an Application

Task Manager is often used to stop or end an application that has crashed and has stopped responding to user input.

TO USE TASK MANAGER TO STOP OR END AN APPLICATION OR PROCESS, FOLLOW THESE STEPS:

1. Start Task Manager (press Ctrl + Alt + Delete, and then click the Task Manager command button in the Windows NT Security dialog box).
2. The Windows NT Task Manager dialog box appears, as shown in Figure 21-3. Notice that the Applications tab is on top, by default. Highlight the application you want to stop. Click the End Task command button. Exit Task Manager.

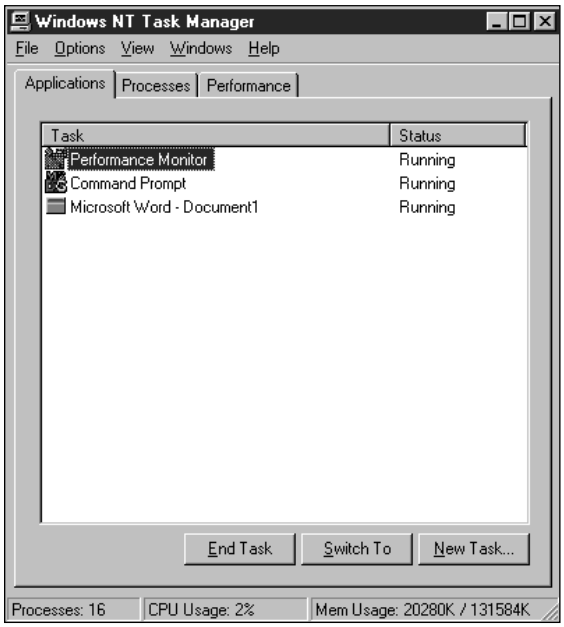


FIGURE 21-3 **Using Task Manager to stop (end) an application**

Using Task Manager to View Performance Statistics

You can also use Task Manager to view performance statistics for your Windows NT computer.

TO USE TASK MANAGER TO VIEW STATISTICS, FOLLOW THESE STEPS:

1. Start Task Manager.
2. The Windows NT Task Manager dialog box appears. Click the Performance tab.
3. The Performance tab appears, as shown in Figure 21-4. Notice the CPU Usage and Memory Usage graphs. When you are finished viewing performance statistics, exit Task Manager.

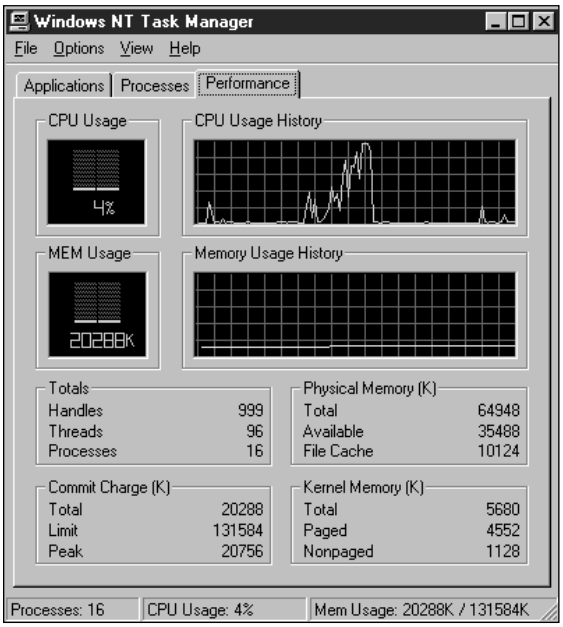


FIGURE 21-4 **Viewing performance statistics in Task Manager**

Using Task Manager to Change an Application's Base Priority

Task Manager can also be used to view individual application and process statistics, and to set an application or process's base priority.

TO VIEW PROCESS STATISTICS OR TO CHANGE AN APPLICATION OR PROCESS'S BASE PRIORITY BY USING TASK MANAGER, FOLLOW THESE STEPS:

1. Start Task Manager.
2. The Windows NT Task Manager dialog box appears. Click the Processes tab.
3. The Processes tab appears, as shown in Figure 21-5. Notice the various statistics displayed. Maximize the Task Manager dialog box. Select View ➪ Select Columns.
4. The Select Columns dialog box appears, as shown in Figure 21-6. Note the various process statistics that you can choose to display in Task Manager. Select the check box next to Base Priority. Click OK.

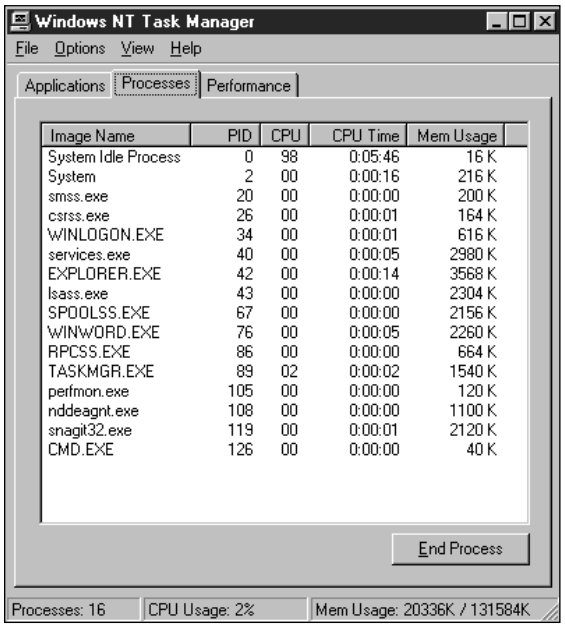


FIGURE 21-5 **Using Task Manager to view process statistics**

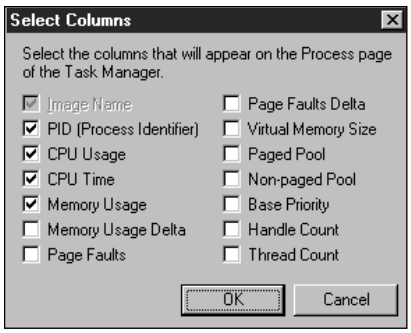


FIGURE 21-6 **Selecting process information to be displayed in Task Manager**

5. The Processes tab in the Windows NT Task Manager dialog box reappears, as shown in Figure 21-7. Note the new column that displays the base priority for each process. Also note that `perfmon.exe` (Performance Monitor) currently has a base priority of High.

   Right-click the application or process for which you want to change the base priority. From the menu that appears, select the base priority (Realtime, High, Normal, or Low) that you want to assign to the application or process.

   Figure 21-8 shows the process of selecting menu options to assign a base priority of Realtime to the `perfmon.exe` (Performance Monitor) process.

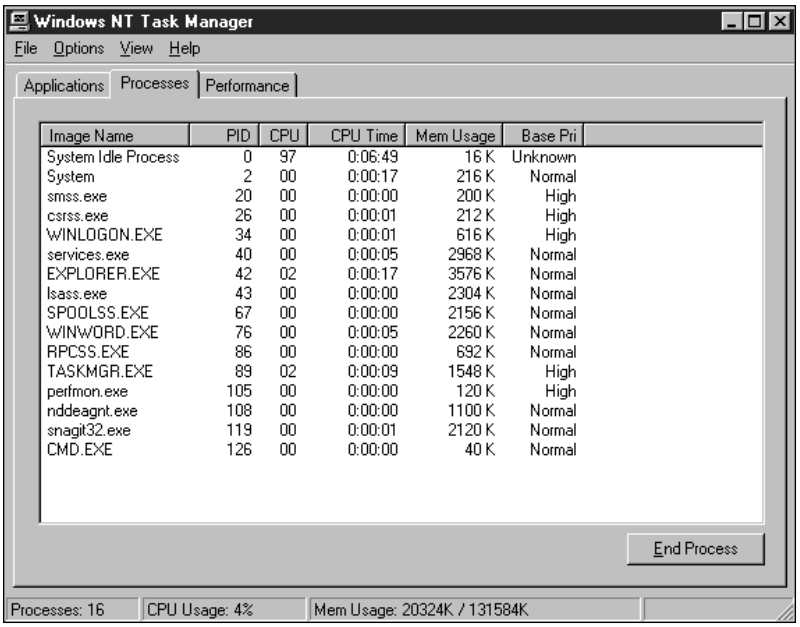


FIGURE 21-7 **Viewing a process's base priority in Task Manager**

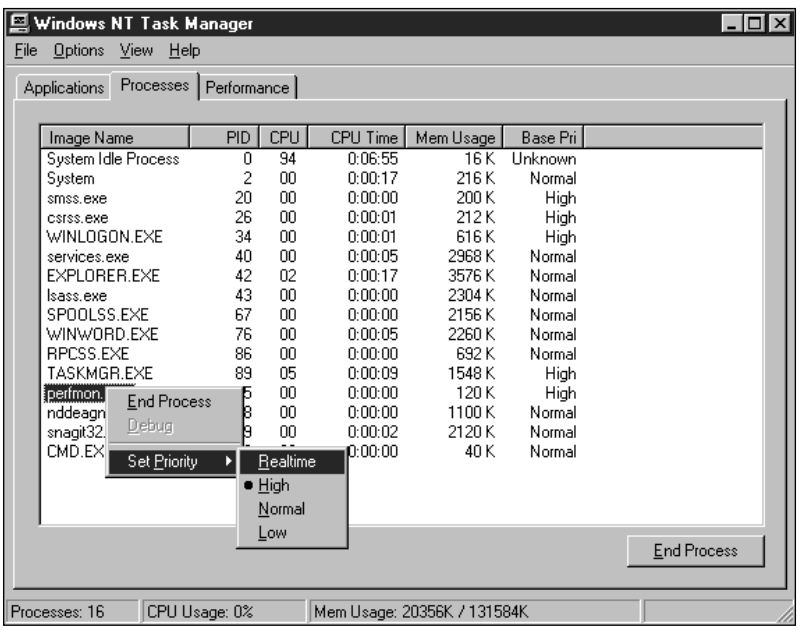


FIGURE 21-8 **Changing a process's base priority in Task Manager**

6. A Task Manager Warning dialog box appears, indicating that changing the base priority of a process may have undesirable results. Click the Yes command button to continue.
7. The Processes tab in the Windows NT Task Manager dialog box reappears, as shown in Figure 21-9. Notice that `perfmon.exe` (Performance Monitor) now has a base priority of Realtime. Exit Task Manager.

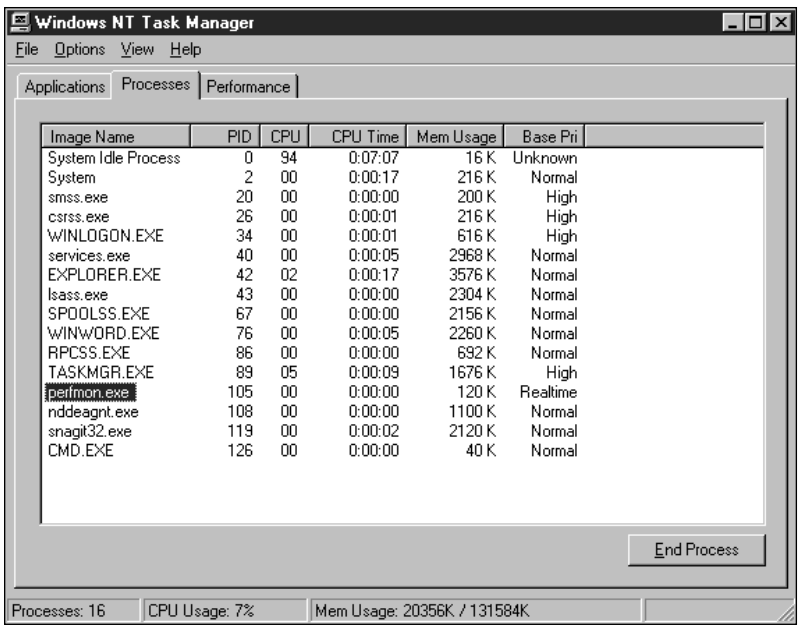


FIGURE 21-9 **Base priority of `perfmon.exe` (Performance Monitor) changed**

Troubleshooting Common Application Problems

Occasionally you may experience problems when running an application on Windows NT. Some application problems have workarounds, and others can't be solved easily because they involve applications that simply aren't compatible with Windows NT.

Table 21-3 shows some common applications problems and recommended solutions.

TABLE 21-3 TROUBLESHOOTING APPLICATION PROBLEMS IN WINDOWS NT

PROBLEM	*RECOMMENDED SOLUTION*
An application crashes and no longer responds to user input.	Use Task Manager to stop (end) the application. (This is sometimes called killing the application.)
A 16-bit Windows (Win16) application often crashes, which results in all other Win16 applications locking up.	Use Task Manager to stop (end) all of the Win16 applications that are running. Configure the application that often crashes to run in its own separate memory space.
An application that makes a direct call to the hardware does not function properly on your Windows NT computer.	Windows NT does not support applications that make direct calls to hardware. Run this application on the operating system for which it was originally designed, or rewrite the application to function correctly in a Windows NT environment.
An OS/2 or POSIX application makes calls to APIs that aren't supported by Windows NT. The application doesn't function properly on your Windows NT computer.	Windows NT does not support calls to OS/2 Presentation Manager APIs (unless you have the special add-on product that supports Presentation Manager), and does not support many POSIX extensions. Run this application on the platform for which it was originally designed, or rewrite the application to function correctly in a Windows NT environment.
A standard Win32 application won't run on your PowerPC (or DEC Alpha, or MIPS R4000) computer.	Purchase a PowerPC (or DEC Alpha, or MIPS R4000) version of the application, or recompile the application for the PowerPC (or DEC Alpha, or MIPS R4000) platform. Win32 applications are source-compatible, and must be recompiled for each individual hardware platform.

Key Point Summary

This chapter explored the various application environments supported by Windows NT 4.0, as well as some techniques for customizing application performance. Windows NT 4.0 is designed to run applications created for several different types

of operating system environments. Windows NT supports these different application types via multiple environment subsystems.

- The application types and operating system environments supported by Windows NT 4.0 include:
  - MS-DOS applications (MS-DOS environment)
  - 16-bit Windows applications, such as those written for Windows 3.*x* and Windows for Workgroups (Win16 environment)
  - 32-bit Windows applications, such as those written for Windows NT and Windows 95 (Win32 environment)
  - POSIX applications (POSIX environment)
  - OS/2 applications, such as those written for OS/2 1.*x* (OS/2 environment)
- *MS-DOS applications* are supported in Windows NT 4.0 by a subsystem called an *NT Virtual DOS Machine (NTVDM)*. An NTVDM emulates an Intel 486 computer running MS-DOS. Each MS-DOS application runs in its own separate NTVDM. Because of this, if an MS-DOS application crashes, other applications are *not* affected. Windows NT can preemptively multitask multiple MS-DOS applications.
- NTVDMs have three threads, although only *one* of these threads is used by the application. A *thread* is the smallest unit of processing that can be scheduled by the Windows NT Schedule service.
- You can edit the `Autoexec.nt` and/or `Config.nt` files to include any necessary MS-DOS environmental settings that would normally be configured in the MS-DOS computer's `Autoexec.bat` or `Config.sys` files.
- *Win16 applications* are 16-bit Windows applications designed for Windows 3.*x* and Windows for Workgroups. Windows NT 4.0 provides support for 16-bit Windows applications via a subsystem called WOW (short for Win16-on-Win32). WOW emulates an Intel 486 computer running MS-DOS *and* Windows 3.1. The WOW subsystem runs in an NTVDM called the Win16 NTVDM.
- By default, when multiple Win16 applications are run at the same time, they all run in a single Win16 NTVDM. This means that, by default, all Win16 applications share the same memory space and the Win16 NTVDM's single thread. Because applications share the same memory space, if one application crashes, other Win16 applications may also crash. Because

multiple Win16 applications share an NTVDM's single thread, Windows NT *can't* preemptively multitask multiple Win16 applications.

- To prevent a rogue Win16 application from crashing other Win16 applications, Windows NT permits a Win16 application to be configured to *run in its own separate memory space.* The advantages of running Win16 applications in separate memory spaces are (1) when an application crashes, other applications are not affected, and (2) Windows NT can preemptively multitask Win16 applications when they are configured to run in separate memory spaces. Disadvantages of running Win16 applications in separate memory spaces are (1) more RAM and system resources are used in this configuration, and (2) some Win16 applications may not work correctly when run in separate memory spaces.
  - There are two methods you can use to run a Win16 application in a separate memory space: You can configure the properties of the shortcut to the application (this is the easiest and most commonly used method); or you can use a batch file that uses the `Start/separate` command.
- The *Win32 environment* is Windows NT's native application environment, and is supported on Windows NT via the Win32 subsystem. Win32 applications are 32-bit Windows applications written for Windows NT and/or Windows 95. Win32 applications are *source-compatible* across all supported hardware platforms, which means that Win32 applications must be recompiled for each hardware platform in order to be run on that platform. Each Win32 application runs in its own separate memory space. Because of this, if a Win32 application crashes, other applications are *not* affected. Windows NT can preemptively multitask multiple Win32 applications.
- *POSIX environment* applications that are written to meet POSIX standards are called POSIX-compliant applications. Windows NT provides support for POSIX-compliant applications via the POSIX subsystem. To fully support POSIX-compliant applications, at least one NTFS partition is required on the Windows NT computer. POSIX applications are source-compatible across all supported hardware platforms, which means that POSIX applications must be recompiled for each hardware platform in order to run on that platform. Each POSIX application runs in its own separate memory space. If a POSIX application crashes, other applications are *not* affected. Windows NT can preemptively multitask POSIX applications.

- *OS/2 environment* applications consist of 16-bit character-based applications designed for OS/2 version 1.*x*. Presentation Manager and OS/2 2.*x* and 3.*x* applications are *not* supported by Windows NT. Windows NT only supports the OS/2 subsystem on Intel 486 and higher platforms. Each OS/2 application runs in its own separate memory space. If an OS/2 application crashes, other applications are *not* affected. Windows NT can preemptively multitask OS/2 applications.
- Windows NT 4.0 supports several different types of applications on various hardware platforms. Supported hardware platforms include the Intel 486 processor, the DEC Alpha processor, the PowerPC processor, and the MIPS R4000 processor.
- An *application priority* is a number between 0 and 31 that is assigned to an application when it is started. Applications that have a high priority receive more processor time than applications with a low priority. Most user applications are assigned a priority of 8, the normal priority. In Windows NT, the `Start` command is used to start applications at various priorities. The six switches commonly used with the `Start` command are: `/low`, `/normal`, `/high`, `/realtime`, `/min`, and `/separate`. The `Start` command can be used in batch files and from the command prompt.
- You can configure the *foreground application priority boost* by zero, one, or two points by using the System application in Control Panel.
- *Windows NT Task Manager* can be used to start and stop (end) applications, to view performance statistics, and to change a process's base priority. There are four different ways to access Task Manager:
  - By pressing Ctrl + Shift + Esc
  - By pressing Ctrl + Alt + Delete, and then clicking the Task Manager command button in the Windows NT Security dialog box.
  - By right-clicking a blank space on the taskbar (on the Windows NT desktop), and then selecting Task Manager from the menu that appears.
  - By selecting Start ➢ Run, and then typing **taskmgr** in the Run dialog box.
- Occasionally you may experience problems when running an application on Windows NT. Some application problems have workarounds, and others can't be solved easily because they involve applications that simply aren't compatible with Windows NT.

Applying What You've Learned

Now it's time to regroup, review, and apply what you've learned in this chapter.

The following Instant Assessment questions bring to mind key facts and concepts. In addition, the review activity provides an opportunity to test your troubleshooting skills.

The hands-on lab exercise will really reinforce what you've learned, and give you a chance to practice some of the tasks tested by the Workstation exam.

Instant Assessment

1. What subsystem in Windows NT 4.0 supports MS-DOS applications?
2. Describe the concept of preemptive multitasking as it applies to Windows NT.
3. What is a thread?
4. Which two files can you edit on a Windows NT computer to include any necessary MS-DOS environmental settings that would normally be configured in the MS-DOS computer's `Autoexec.bat` and/or `Config.sys` files?
5. By default, when multiple Win16 applications are run at the same time on a Windows NT computer, where do they run?
6. What can you do to prevent a rogue Win16 application from crashing other Win16 applications?
7. What is the `Start/separate` command used for?
8. What is the native Windows NT application environment?
9. What does the phrase "source-compatible across all hardware platforms" mean?
10. What is required on a Windows NT computer to fully support POSIX-compliant applications?
11. What number represents a normal Windows NT user application priority?
12. What would you type at the command prompt on a Windows NT computer to start Notepad in a minimized window with an application priority of 13 (high).
13. What Windows NT application can you use to change the foreground application priority boost?

14. What are the four ways you can access Windows NT Task Manager?

15. What kinds of applications does the Windows NT OS/2 environment support?

T/F

16. On Windows NT computers, each MS-DOS application runs in its own separate NTVDM. ___

17. Each Win32 application runs in its own separate memory space. ___

18. If one Win32 application crashes, all other Win32 applications also crash. ___

19. Win32 applications and POSIX applications are *not* source-compatible across all supported hardware platforms on Windows NT. ___

For answers to the Instant Assessment questions see Appendix D.

Review Activity

Workstation

The following activity gives you a chance to apply your knowledge of running applications on Windows NT computers in two real-life troubleshooting situations.

Troubleshooting common application problems

The purpose of this exercise is to give you experience in troubleshooting common application problems on Windows NT computers. For each of the following problems, recommend a solution.

Problem 1 An application has crashed on your Windows NT computer, and won't respond to any input from the mouse or keyboard. What should you do?

Problem 2 You try to run a POSIX-compliant application compiled for a DEC Alpha processor on your Intel Pentium Pro/200MHz computer that is running Windows NT, but the application won't run properly. What should you do?

For answers to the Review Activity see Appendix D.

Hands-on Lab Exercise

The following hands-on lab exercise provides you with a practical opportunity to apply the knowledge you've gained in this chapter.

Lab 21.34 *Starting applications at various priorities*

Workstation

The purpose of this lab exercise is to give you hands-on experience in configuring foreground application responsiveness and in starting applications at various priorities on a Windows NT Workstation computer.

This lab consists of two parts:

Part 1: Using the System application to configure foreground application responsiveness

Part 2: Using the `Start` command and Task Manager to start applications at various priorities

Begin this lab by booting your computer to Windows NT Workstation. Log on as Administrator. Place your Windows NT Workstation compact disc in your computer's CD-ROM drive.

Follow the steps below carefully.

Part 1: Using the System application to configure foreground application responsiveness

In this section, you use the System application in Control Panel to change foreground application responsiveness on a Windows NT Workstation computer. Additionally, you install games on your computer (if you haven't already done so).

1. Select Start ➪ Settings ➪ Control Panel.
2. The Control Panel dialog box appears. Double-click the System icon.
3. The System Properties dialog box appears. Click the Performance tab.
4. The Performance tab appears. Notice the Application Performance section on this tab. Move the Boost slide bar to the middle, halfway between None and Maximum. (This configures a performance boost of one point for the foreground application.) Move the Boost slide bar back to Maximum. (This configures a performance boost of 2 points for the foreground application.) Click OK.
5. The Control Panel dialog box reappears. If you have already installed games (including Pinball) on your Windows NT Workstation computer, close Control Panel, skip the remaining steps in this section, and continue

on to Part 2. If you haven't yet installed games on your Windows NT Workstation computer, double-click the Add/Remove Programs icon, and continue on to step 6.

6. The Add/Remove Programs Properties dialog box appears. Click the Windows NT Setup tab.
7. The Windows NT Setup tab appears. Select the check box next to Games in the Components list box. Click OK.
8. The Add/Remove Programs Properties—Copying Files dialog box appears. If the Files Needed dialog box appears, ensure that the correct path to the source files (usually the `I386` folder on your Windows NT Workstation compact disc) is presented in the "Copy files from" text box. Edit this text box as necessary. Click OK.
9. Windows NT Workstation copies files.
10. The Control Panel dialog box reappears. Close Control Panel.

    Continue on to Part 2.

Part 2: Using the Start command and Task Manager to start applications at various priorities

In this section, you use the `Start` command and Windows NT Task Manager to start applications at various priorities on a Windows NT Workstation computer.

1. Select Start ➪ Programs ➪ Command Prompt.
2. The Command Prompt dialog box appears. At the command prompt, type **cd \winntwks\system32** and press Enter.
3. At the `C:\WINNTWKS\system32>` command prompt, type **start /low musrmgr.exe** and press Enter.
4. User Manager starts. Press Ctrl + Alt + Delete.
5. The Windows NT Security dialog box appears. Click the Task Manager command button.
6. The Windows NT Task Manager dialog box appears. Click the Processes tab.
7. The Processes tab appears. Select View ➪ Select Columns.
8. The Select Columns dialog box appears. Deselect the check boxes next to PID (Process Identifier) and CPU Usage. Select the check box next to Base Priority. Click OK.
9. The Processes tab in the Windows NT Task Manager dialog box reappears. Find `USRMGR.EXE` in the list box. Notice that `USRMGR.EXE` has a base priority of Low. Click the Applications tab.

10. The Applications tab appears. Highlight User Manager and click the End Task command button at the bottom of the dialog box. (This action stops User Manager). Minimize (*don't* close) Windows NT Task Manager.
11. At the `C:\WINNTWKS\system32>` command prompt, type **start /realtime usrmgr.exe** and press Enter.
12. User Manager starts. Maximize Task Manager, and click the Processes tab.
13. The Processes tab appears. Notice that `USRMGR.EXE` now has a base priority of Realtime. Highlight `USRMGR.EXE` and click the End Process command button.
14. A Task Manager Warning dialog box appears. Click the Yes command button. (This action stops User Manager.)
15. Minimize (*don't* close) Windows NT Task Manager.
16. At the `C:\WINNTWKS\system32>` command prompt, type **start /realtime perfmon.exe** and press Enter.
17. Performance Monitor starts. Maximize Task Manager.
18. The Processes tab appears. Notice that `perfmon.exe` has a base priority of High. (The `/realtime` switch does *not* always work with all applications. Performance Monitor always starts with a base priority of High, no matter which priority is assigned with the `Start` command.)

    Right-click `perfmon.exe`. Select Set Priority ⩾ Realtime from the menu that appears.
19. A Task Manager Warning dialog box appears. Click the Yes command button.
20. In the Windows NT Task Manager dialog box, notice that `perfmon.exe` now has a base priority of Realtime. (This is how you can change the priority of applications that don't respond correctly to the priority setting specified by the `Start` command.)

    Highlight `perfmon.exe` and click the End Process command button.
21. A Task Manager Warning dialog box appears. Click the Yes command button.
22. Minimize (*don't* close) Windows NT Task Manager.
23. At the `C:\WINNTWKS\system32>` command prompt, type **cd "\program files\windows nt\pinball"** and press Enter.
24. At the `C:\Program Files\Windows NT\PINBALL>` command prompt, type **start /realtime pinball** and press Enter.

25. The Pinball game starts. Notice that your computer is locked up and won't respond to keyboard or mouse input. As you can see, not all programs should be run at real-time priority. In this case, when Pinball is set at real-time priority, it utilizes all available system resources, so that even the operating system is unable to function properly. Press the Restart button on your computer, or power off your computer.