chapter 15

Windows Under the Hood

"Something's wrong with my CPU. My Register must be off because Windows is all blue and stuff and I can't log in. I need to get to my Internet. Please help."

—Anonymous tech call

In this chapter, you will learn how to

- **Work with the Registry**
- **Understand and observe in detail the Windows boot process**
- **Control processes and services**
- **Explore Windows tools for programmers**

Windows is powerful, easy to use, surprisingly idiot proof, backward compatible, and robust. A large part of Windows' power is hidden—*under the hood*—in programs and processes that Microsoft doesn't want normal users to see. For the record, I think hiding anything that normal users don't need to access is a smart idea; they can't break what they can't find. Technicians, on the other hand, need to not only understand these processes and programs but also know how to use them, configure them, and fix them when needed. Let's start with one of the most famous and most important items under the hood: the Registry.

802

Registry

The **Registry** is a huge database that stores everything about your PC, including information on all of the hardware in the PC, network information, user preferences, file types, and virtually anything else you might run into with Windows. Almost any form of configuration you do to a Windows system involves editing the Registry. Every version of Windows stores the numerous Registry files (called *hives*) in the \%SystemRoot%\System32\ config folder and each user account folder. Fortunately, you rarely have to access these massive files directly. Instead, you can use a set of relatively tech-friendly applications to edit the Registry.

The CompTIA A+ 220-802 certification exam does not expect you to memorize every aspect of the Windows Registry. You should, however, understand the basic components of the Registry, know how to edit the Registry manually, and know the best way to locate a particular setting.

Accessing the Registry

Before you look in the Registry, let's look at how you access the Registry directly by using a Registry editor. Once you know that, you can open the Registry on your machine and compare what you see to the examples in this chapter.

Before Windows XP, Windows came with two Registry editors: regedt32.exe and the much older regedit.exe. You started either of these programs by going to a command prompt and typing its filename.

The reason for having two different Registry editors is long and boring, and explaining it would require a very dull 15-minute monologue about how the Registry worked in Windows 9*x* and Windows NT. Suffice it to say that starting with Windows XP, Microsoft eliminated the entire two-Registry-editor nonsense by creating a new regedt32 that includes strong search functions. No longer are there two separate programs, but interestingly, entering either regedit or regedt32 at a command prompt (or in the Start | Run dialog box or Start | Search bar) brings up the same program, so feel free to use either program name. You may also dispense with calling the Registry Editor by its filename and use its proper title.

You might see either regedit or regedt32 on the exam as the way to access the Registry Editor.

Registry Components

The Registry is organized in a tree structure similar to the folders in the PC. Once you open the Registry Editor in Windows, you will see five main subgroups, or **root keys**:

- HKEY_CLASSES_ROOT
- HKEY_CURRENT_USER
- HKEY_USERS

- HKEY_LOCAL_MACHINE
- HKEY_CURRENT_CONFIG

Try opening one of these root keys by clicking on the plus sign to its left; note that more subkeys are listed underneath. A subkey also can have other subkeys, or *values*. Values define aspects of the subkey. Figure 15.1 shows an example of a subkey with some values. Notice that the Registry Editor shows only keys—root keys and subkeys—on the left and values on the right. Each of the root keys has a specific function, so let's take a look at them individually.

HKEY_CLASSES_ROOT

This root key defines the standard *class objects* used by Windows. A class object is a named group of functions that defines what you can do with the object it represents. Pretty much everything that has to do with files on the system is defined by a class object. For example, the Registry uses two class objects to define the popular MP3 sound file.

If you search my computer's Registry for the .PDF file extension, you will find a class object that associates the .PDF file extension with the name "FoxitReader.Document" (see Figure 15.2). Unless you use the popular Foxit Reader for your PDFs, you'll see a different program name, but the process still works.

So what are the properties of FoxitReader.Document? That's what the HKEY_CLASSES_ROOT root key is designed to handle. Search this section again for "FoxitReader.Document" (or whatever it said in the value for your PDF file) and look for a subkey called "open." This subkey tells the system everything it needs to know about a particular software item, from

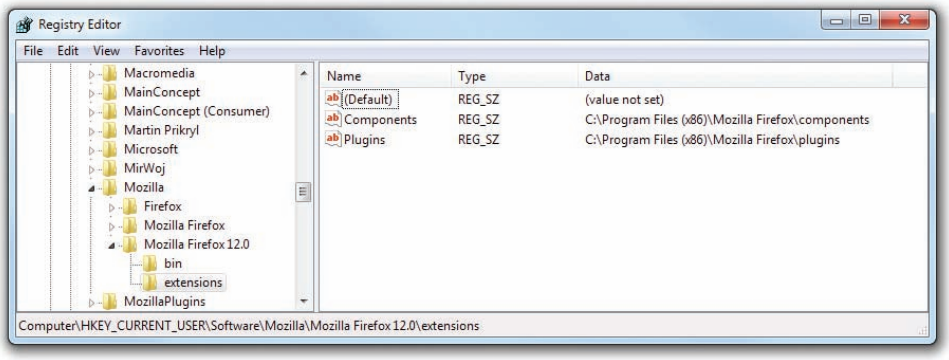


Figure 15.1 Typical Registry root keys, subkeys, and values

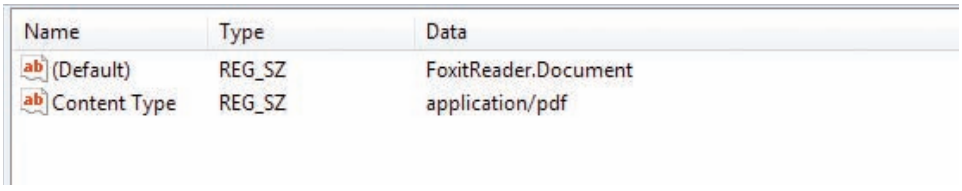
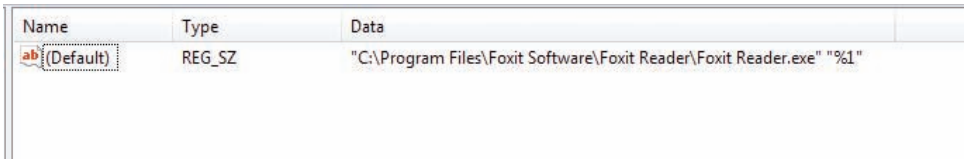


Figure 15.2 Association of .PDF with Foxit Reader

which program to use to open a file (its **file association**), to the type of icon used to show the file, to what to show when you right-click on that file type (see Figure 15.3).

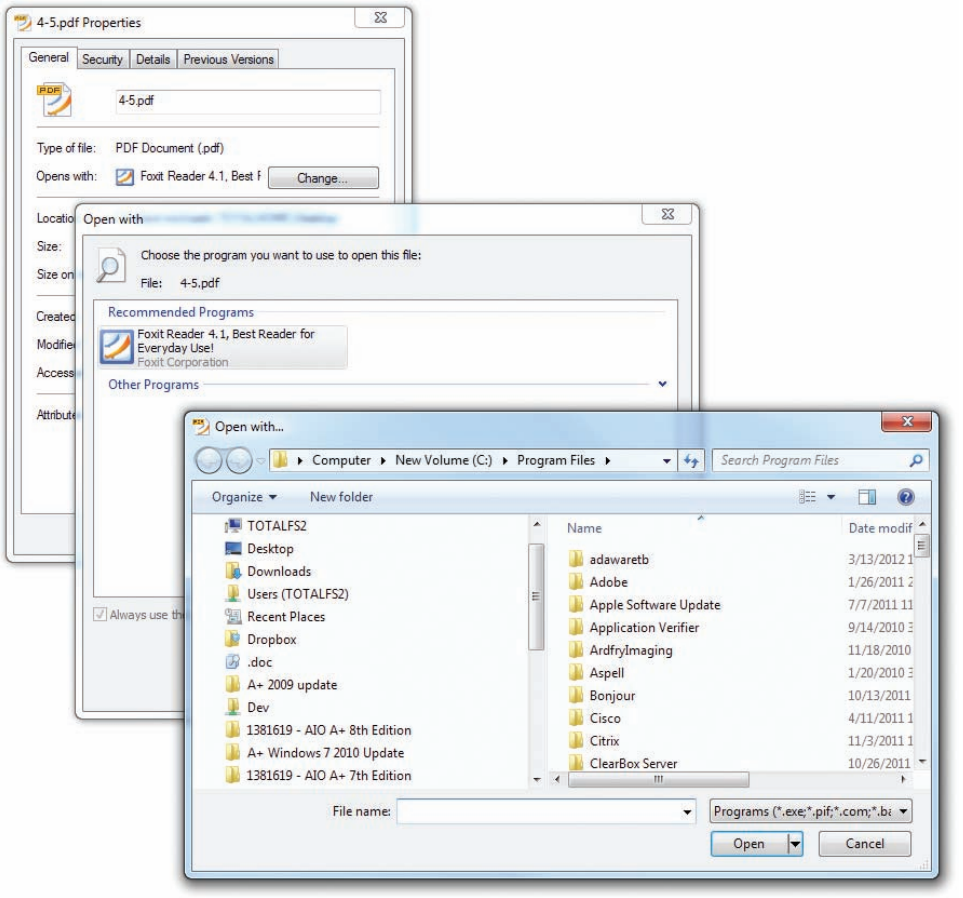


• **Figure 15.3** PDF file settings

Although it is possible to change most of these settings in the Registry Editor, the normal way is to choose more user-friendly methods. Select a data file, right-click on the data file, and click Properties. Click the Change button on the General tab to open the Open with dialog box. From there, you can select the program you want to use. If Windows knows of another program designed for that file type, it will show you these alternative programs. If not, you can simply browse for the program you want to use (see Figure 15.4).

HKEY_CURRENT_USER and HKEY_USERS

Windows is designed to support more than one user on the same PC, storing personalized information such as desktop colors, screensavers, and the



• **Figure 15.4** Changing the file association the easy way

contents of the desktop for every user that has an account on the system. HKEY_CURRENT_USER stores the current user settings, and HKEY_USERS stores all of the personalized information for all users on a PC. While you certainly can change items such as the screensaver here, the better way is to right-click on the desktop and select Properties.

HKEY_LOCAL_MACHINE

This root key contains all the data for a system's non-user-specific configurations. This encompasses every device and every program in your PC.

HKEY_CURRENT_CONFIG

If the values in HKEY_LOCAL_MACHINE have more than one option, such as two different monitors, this root key defines which one is currently being used. Because most people have only one type of monitor and similar equipment, this area is almost never touched.

Talkin' Registry

When describing a Registry setting, we use a simple nomenclature. For example, I recently moved my copy of *World of Warcraft* from my C: drive to my D: drive and was having problems when the program started. I went online to www.blizzard.com (home of Blizzard Entertainment, the folks who make *World of Warcraft*) and contacted the support staff, who gave me instructions to access the Registry and make this change:

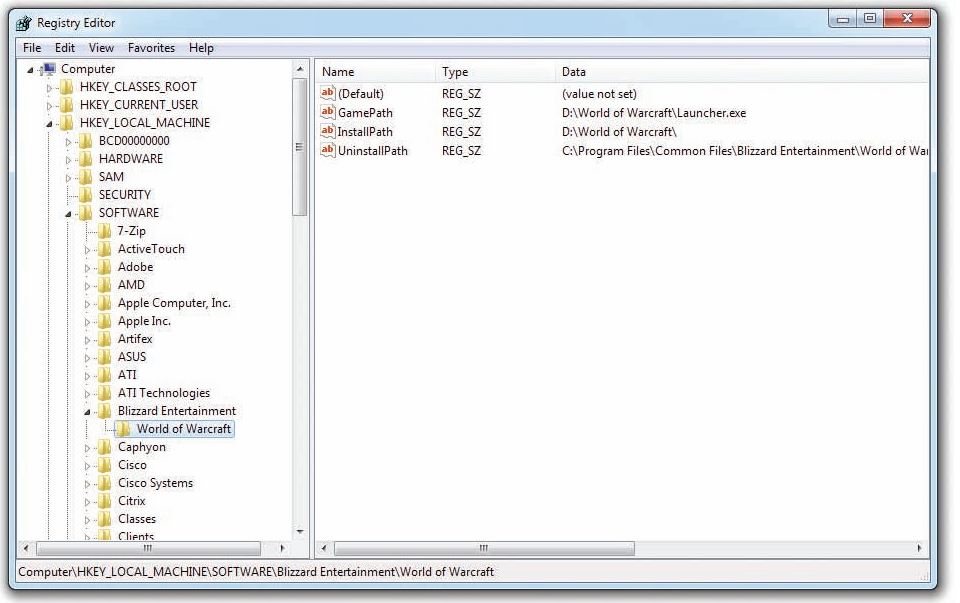
> "Go to HKLM\SOFTWARE\Blizzard Technologies\World of Warcraft and change the GamePath object and the InstallPath object to reflect the new drive letter of your new WoW location."

To do so, I fired up regedit. Using this nomenclature, I was able to find the location of these Registry settings. Figure 15.5 shows this location. Compare this image to the path described in the instructions from Blizzard. Note that HKEY_LOCAL_MACHINE is abbreviated as HKLM. You will see this abbreviation on the exam!

To describe the location of a specific Registry value, like where the Blizzard tech told me to go, requires a little bit of repetition. To wit, in the previous example, World of Warcraft is a subkey to Blizzard Technologies, which is in turn a subkey to the root key HKLM. The World of Warcraft subkey has four values. All keys have the (Default) value, so in this case the World of Warcraft subkey offers three functional values.

Values must have a defined type of data they store:

- **String value** These are the most flexible type of value and are very common. You can put any form of data in these.
- **Binary value** These values store nothing more than long strings of ones and zeros.
- **DWORD value** These values are like Binary values but are limited to exactly 32 bits.
- **QWORD value** These values are like Binary values but are limited to exactly 64 bits.



• **Figure 15.5** Editing the Registry to move World of Warcraft to a new drive

There are other types of values, but these four are used for 98 percent of all Registry entries.

Manual Registry Edits

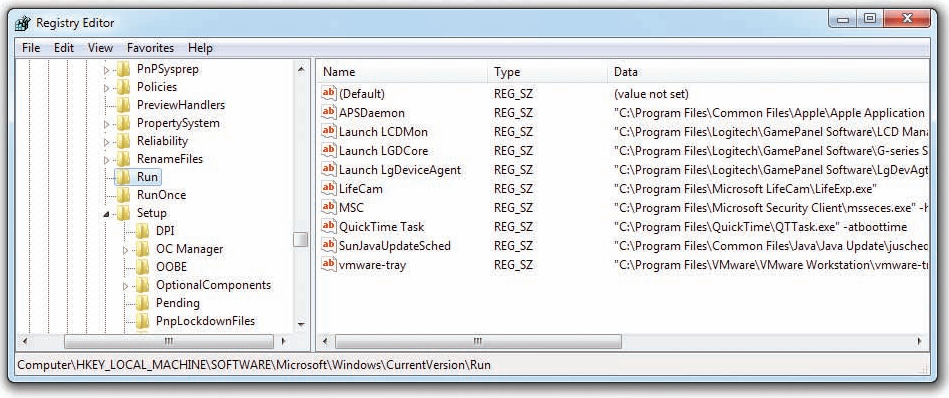
There's little motivation for you to go into the Registry and make manual edits unless you've done some research that tells you to do so. When you do find yourself using the Registry Editor to access the Registry, you risk breaking things in Windows: applications might not start, utilities might not work, or worst of all, your computer might not boot. To prevent these problems, always make a backup of the Registry before you change anything. Once the backup is in a safe place (I like to use a thumb drive, personally), reboot the system to see if the changes you made had the desired result. If it worked, great. If not, you'll need to restore the old Registry settings using your backup. Let's watch this in action.

One of the more common manual Registry edits is to delete autostarting programs. I want to prevent three programs installed by my Logitech GamePanel keyboard and mouse from autostarting. The most common place for making this change is here:

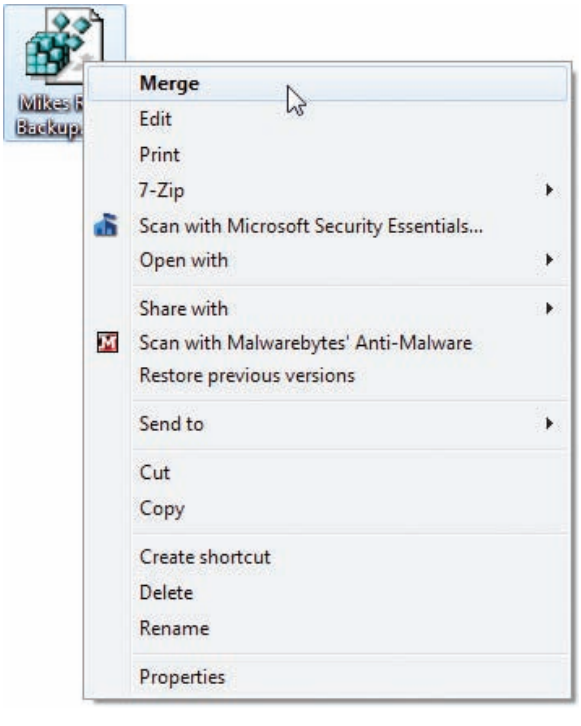
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

Opening the Registry Editor and going to this subkey, you'll see something like Figure 15.6.

Before I delete these keys, I'm going to save a copy of my Registry. The Registry Editor's Export feature enables you to save either the full Registry or only a single root key or subkey (with all subkeys and values under it). Select Run from the left pane and then click File | Export. Save the subkey as a Registration file with the extension .reg. Be sure to put that file somewhere you'll remember. Should you need to restore that key, use the File |



• **Figure 15.6** Mike's Run subkey



• **Figure 15.7** Merging keys from a backup file

Import command, or just right-click on the icon as shown in Figure 15.7 and click Merge.

Command-line Registry Editing Tools

Windows includes a couple of command-line tools to edit the Registry (plus a lot more in the PowerShell). The two that you might need on occasion are reg and regsrv32.

The reg command is a full Registry editing tool. You can view Registry keys and values, import and export some or all of a Registry, and even compare two different versions of a Registry. The tool is so powerful that it has multiple levels of help so you can tailor a command to accomplish very tight Registry edits. For example, typing **reg /?** brings up a list of 12 specific operations that you can search for help on, such as reg query /? and reg add /?.

The regsrv32 command, in contrast with reg, can modify the Registry in only one way, adding (or *registering*) dynamic link library (DLL) files as command components. For the scoop on what regsrv32 does, please see "Component Services" later in the chapter for a description of the graphical tool that does the same thing. Only the 32-bit versions of Windows have the regsrv32 command, by the way.

If the command-line interface is new to you, you might want to flag this section of Chapter 15 and skip it for now, then return to it after reading about the command line and how it works in Chapter 18.

The Boot Process

The Windows installation creates a number of specific files and folders that the OS needs to run a PC. Some of these files and folders are directly on the root of the C: drive; others can be elsewhere. The best way to remember the locations of these files and folders and to know their importance to the OS is by looking at how they interact to boot the PC.

Booting Windows XP differs dramatically from booting Windows Vista/7. Windows XP was the last Microsoft operating system to use a very old boot process known as ntldr (NT Loader). Windows Vista introduced

the far more flexible, complex, and powerful Windows Boot Manager (or **bootmgr**), which Windows 7 also uses. The great difference between these two boot managers requires us to look at both in more depth.

The Windows XP Boot Process

Windows XP distinguishes between the files that start the operating system (called the *system files*) and the rest of the operating system files (by default in the \Windows folders). The system files (memorize these!) consist of three required files: ntldr, boot.ini, and ntdetect.com. If you're using a SCSI hard drive, there's a fourth file called ntbootdd.sys. The ntldr (pronounced *NT loader*) file begins the boot process.

To see these files, go into My Computer and open the C: drive. Go to Tools | Folder Options. Click *Show hidden files and folders*, uncheck the *Hide protected operating system files (Recommended)* option, and click OK. Now when you return to viewing the folder in My Computer, you will see certain critical files that Windows otherwise hides from you to prevent you from accidentally moving, deleting, or changing them in some unintended way.

You know from Chapter 12 that to make a drive bootable requires an active, primary partition, right? Let's look at the process in a PC with a hard drive partitioned as C: and D:.

The CPU wakes up and runs the system BIOS, and then the BIOS sends out a routine that looks on the first bootable drive's partition table for a boot sector marked as active. This is usually the C: drive. The boot sector takes over from BIOS and reads the master file table (MFT) in the bootable partition. The MFT points to the location of the Windows XP system files. Windows XP calls the primary active partition the *system partition* or the *system volume* (if it's a dynamic disk).

We're almost booted, but to complete the process, we need to take a moment to learn about some more critical Windows XP files: the Windows boot files and the Windows system files. The Windows XP *boot files* consist of ntoskrnl.exe (the Windows kernel), the \Winnt\System32\Config\System file (which controls the loading of device drivers), and the device drivers. Although these files are the core of the Windows XP operating system, they are not capable of booting, or starting, the system. For that feat, they require ntldr, ntdetect.com, and boot.ini—the system files.

The system files start the PC and then, at the end of that process, point the CPU to the location of the boot files. The CPU goes over and chats with ntoskrnl, and the GUI starts to load. The operating system is then up and running, and you're able to do work.

The odd part about all of this is that Microsoft decided to make the OS files mobile. The Windows XP operating system files can reside on any partition or volume in the PC. The \Windows folder, for example, could very well be on drive D:, not drive C:. Whichever drive holds the core OS files is called the *boot partition* or *boot volume*. This can lead to a little confusion when you say the system files are on the C: drive and Windows is on the D: drive, but that's just the way it is. Luckily, the vast majority of Windows XP systems have the system partition and the boot partition both on the same big C: partition.

You know the general process now, so let's look more specifically at the makeup and function of the individual files involved in the boot process.

ntldr

When the system boots up, the master boot record (MBR) or MFT on the hard drive starts the ntldr program. The **ntldr** program then launches Windows XP. To find the available OSs, the ntldr program must read the

boot.ini configuration file. To do so, it loads its own minimal file system, which enables it to read the boot.ini file off of the system partition.

boot.ini

The **boot.ini** file is a text file that lists the OSs available to ntldr and tells ntldr where to find the boot partition (where the OS is stored) for each of them. The boot.ini file has sections defined by headings enclosed in brackets. A basic boot.ini file in Windows XP looks like this:

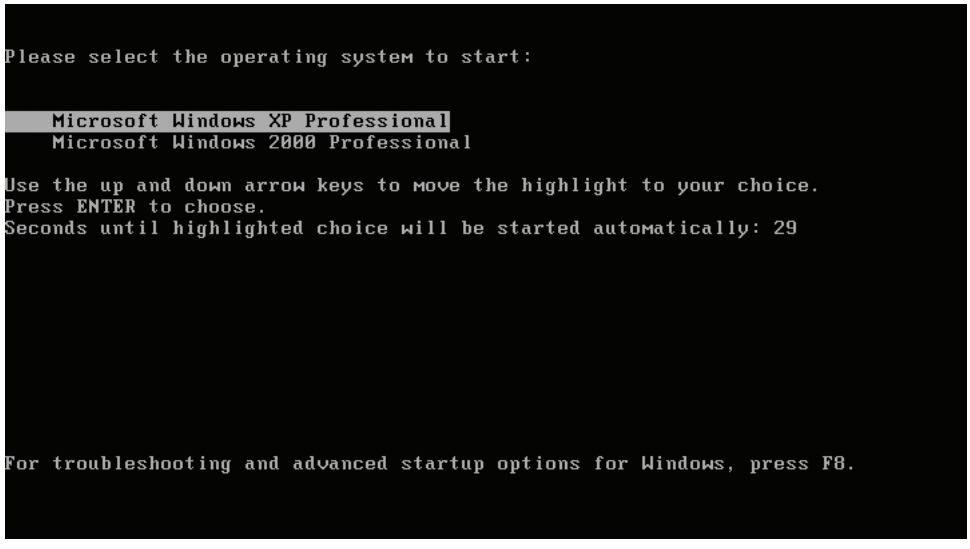
```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)\WINDOWS
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Microsoft
Windows XP Professional" /fastdetect
```

A more complex boot.ini file may look like this:

```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)\WINDOWS
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Microsoft
Windows XP Professional" /fastdetect
multi(0)disk(0)rdisk(0)partition(1)\WINNT="Microsoft Windows
2000 Professional" /fastdetect
```

Such a boot.ini file would result in the boot menu that appears in Figure 15.8.

This crazy `multi(0)disk(0)rdisk(0)partition(1)` is an example of the Advanced RISC Computing (ARC) naming system. It's a system that's designed to enable your PC to boot Windows from any hard drive, including removable devices. Let's take a quick peek at each ARC setting to see how it works.



• **Figure 15.8** Boot loader in Windows XP showing dual-boot configuration

`Multi(x)` is the number of the adapter and always starts with 0. The adapter is determined by the boot order you set in your CMOS setting. For example, if you have one PATA controller and one SATA controller, and you set the system to boot first from the PATA controller, any drive on that controller will get the value `multi(0)` placed in its ARC format. Any SATA drive will get `multi(1)`.

`Disk(x)` is only used for SCSI drives, but the value is required in the ARC format, so with ATA systems it's always set to `disk(0)`.

`Rdisk(x)` specifies the number of the disk on the adapter. On a PATA drive, the master is `rdisk(0)` and the slave is `rdisk(1)`. On SATA drives, the order is usually based on the number of the SATA connection printed on the motherboard, though some systems allow you to change this in CMOS.

`Partition(x)` is the number of the partition or logical drive in an extended partition. The numbering starts at 1, so the first partition is `partition(1)`, the second is `partition(2)`, and so on.

`\WINDOWS` is the name of the folder that holds the boot files. This is important to appreciate! The ARC format looks at the folder, so there's no problem running different editions of Windows XP on a single partition. You can simply install them in different folders. Of course, you have other limitations, such as file system type, but in general, multibooting with Windows is pretty easy. Better yet, this is all handled during the installation process.

You can't boot multiple installations of Windows Vista or Windows 7 from a single partition because their setup programs don't allow you to choose the installation directory.

ARC format can get far more complicated. SCSI drives get a slightly different ARC format. For example, if you installed Windows on a SCSI drive, you might see this ARC setting in your boot.ini file:

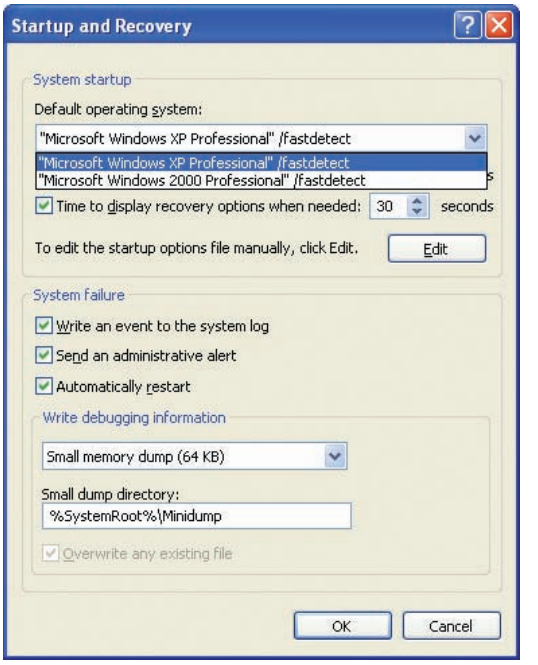
```
scsi(0)disk(1)rdisk(0)partition(1)
```

If you want to boot to a SCSI drive, Windows adds a fourth file to your system files called ntbootdd.sys. This file only exists if you want to boot to a SCSI drive. Most people don't boot to a SCSI, so don't worry if you don't see this file with the other three system files.

On rare occasions, you might find yourself needing to edit the boot.ini file. Any text editor handily edits this file, but most of us prefer to edit boot.ini via the System Setup dialog box. In Windows XP, open the System applet from the Control Panel. Click the Advanced tab and then click the Startup and Recovery button. The boot.ini options show up at the top (see Figure 15.9).

Boot.ini has some interesting switches at the end of the ARC formats that give special instructions on how the operating system should boot. Sometimes Windows puts these in automatically, and sometimes you will add them manually for troubleshooting. Here are a few of the more common ones:

- **/bootlog** Tells Windows to create a log of the boot process and write it to a file called ntbtlog.txt.
- **/cmdcons** Tells Windows to start the Recovery Console (see Chapter 19 for a discussion of the Recovery Console).



• **Figure 15.9** The boot.ini options

- **/lastknowngood** Tells Windows to boot the Last Known Good set of files.
- **/noexecute** Newer CPUs come with Data Execute Protection (DEP) to prevent unruly programs from causing system lockups. The setting for this, /noexecute=optin, is the default on Windows systems.

ntdetect.com

If ntldr determines that you have chosen to start Windows XP, it boots the system into protected mode and then calls on **ntdetect.com** to detect the installed hardware on the system. Ntldr then refers to the boot.ini file to locate the Windows boot files.

Critical Boot Files

The Recovery Console can be used to restore damaged, corrupted, or missing ntldr and ntdetect.com files from the Windows XP installation disc.

Naming all the critical boot files for Windows XP is akin to naming every muscle in the human body—completely possible, but time-consuming and without any real benefit. A few of the *most* important files certainly deserve a short mention.

Once ntldr finishes detections, it loads ntoskrnl.exe, hal.dll, some of the Registry, and some basic device drivers; then it passes control to the ntoskrnl.exe file. Ntoskrnl.exe completes the Registry loading, initializes all device drivers, and starts the winlogon.exe program, which displays the Windows XP logon screen (see Figure 15.10).

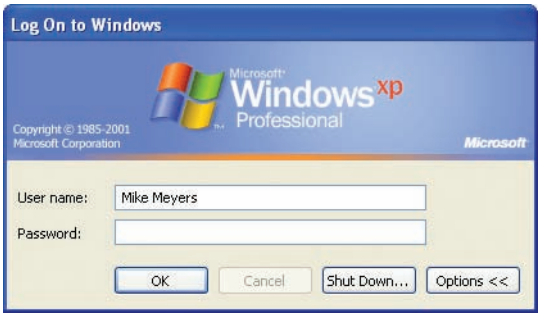


Figure 15.10 Where do you want to go today?

Take the time to memorize the primary boot files and the boot process for Windows XP. Most boot errors are easily repaired if you know which files are used for booting and in which order they load.

The Windows Vista/7 Boot Process

Windows Vista and Windows 7 have a very different boot process than previous versions of Windows. For one thing, Vista/7 supports both BIOS and UEFI, whereas older versions of Windows did not, so things are a bit more complex right off the bat. So not only is there a totally new boot process, this newer Windows Vista/7 boot process varies between two slightly different boot processes: one for systems using BIOS and one for systems with UEFI.

In Windows Vista, bootmgr resides in the root directory of the boot drive. Windows 7, however, keeps bootmgr in the special 100-MB system partition you learned about in Chapter 12. If you are using a UEFI system, the helpfully named EFI system partition contains a special version of bootmgr called bootmgr.efi.

The very first thing that happens when you power on a system with Vista/7 is that either the BIOS or the UEFI starts up. The difference between BIOS and UEFI systems is in what happens next.

- In a BIOS-based system, the BIOS uses its boot order to scan a hard drive for a master boot record (MBR). The MBR holds a small bit of file system boot code that scans the partition table for the system partition and then loads its boot sector. The boot sector in turn contains code that does nothing but point the boot process toward a file called bootmgr (pronounced *Boot Manager*, or "boot mugger" if you're trying to make nerds laugh).

- In a UEFI system, on the other hand, neither the MBR/GUID partition table (GPT) nor the file system boot code is run, and UEFI simply loads bootmgr directly.

If you've ever run a dual-boot system with Vista or 7 as one of the operating systems, you're probably already somewhat familiar with bootmgr; one of its jobs is displaying that "Which operating system do you want to load?" screen and then loading the appropriate operating system. When bootmgr starts, it reads data from a **Boot Configuration Data (BCD)** file that contains information about the various operating systems installed on the system as well as instructions for how to actually load (bootstrap) them. Once an operating system is selected (or immediately if only one is present), bootmgr loads a program called winload.exe, which readies your system to load the operating system kernel itself rather like the way you clean up your house before Aunt Edna comes to visit. It does this by loading into memory the hardware abstraction layer, the system Registry, and the drivers for any boot devices before the operating system itself takes over.

Once the operating system process (called ntoskrnl.exe) takes over, it loads up all of the various processes and systems that comprise Windows, the Windows logo comes up, and you're happily computing, completely oblivious to all of the complex electronic communication that just took place inside your computer.

Tech Tip

Bootmgr Is Missing!

If you use Windows Vista/7 long enough, you may encounter an error message saying that Windows cannot boot because bootmgr is missing. This message is generated when the boot sector code is unable to locate bootmgr, which can be caused by file system corruption, a botched installation, or viruses.

Cross Check

UEFI

You learned about UEFI back in Chapter 8. What are some of the advantages of UEFI? How is it different from BIOS?

Unlike Windows XP, the boot files and the system files must all reside on the same partition in Vista/7.

The BCD file replaces the boot.ini file used in previous operating systems and can be altered by using the command-line tool bcdedit.exe.

For the purposes of the CompTIA A+ 220-802 exam, I'm simplifying things a little, but know that processes, services, and threads can get a lot more complicated.

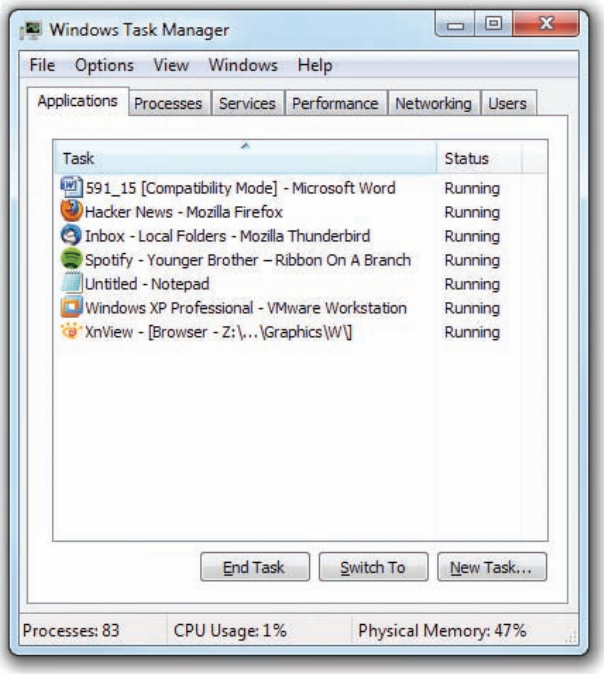
All versions of Windows use services.

Processes and Services and Threads, Oh My!

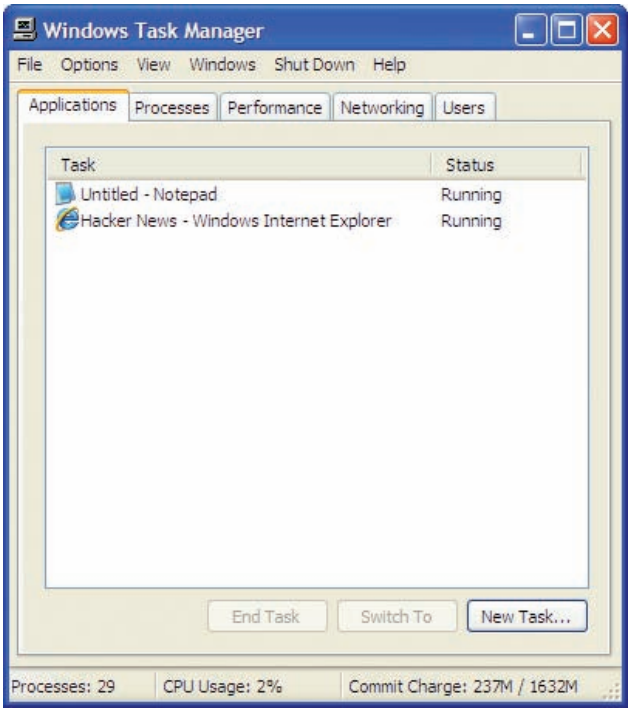
Back in Chapter 6, you learned that CPUs run threads—bits of programs that are fed into the CPU. Let's see how all of this looks from Windows' point of view.

In Windows, programs are executable files waiting on a mass storage device. When you start a program, Windows loads it into RAM as a process. Once there, the CPU reads the process and the process tells the CPU which chunks of code to run. Dealing with processes in their many forms is a big part of understanding what's happening "under the hood."

Windows is a multitasking operating system, running lots of processes simultaneously. Many of these processes appear in a window (or full screen) when you open them and end when you close that window. These processes are called applications. There's also an entire class of processes that, due to the nature of their job, don't require a window of any form. These processes run invisibly in the background, providing a large number of necessary support roles. Collectively, these are called *services*. Let's look at applications, services, and processes and the tools we use to control them.



• **Figure 15.11** Task Manager in Windows 7



• **Figure 15.12** Task Manager in Windows XP

Task Manager

The Windows **Task Manager** is the one-stop-shop for anything you need to do with applications, processes, and, if you're using Windows Vista or Windows 7, services (see Figure 15.11). The quick way to open the Task Manager is to press CTRL-SHIFT-ESC. There are two other ways to open the Task Manager that you might see on the CompTIA A+ exams: go to Start | Run or Start | Search, type **taskmgr**, and press ENTER; or press CTRL-ALT-DELETE and select Task Manager.

If you're running Windows XP, the Task Manager is similar (see Figure 15.12) but lacks a few handy features offered in Windows Vista and Windows 7's Task Manager. I'll cover these differences in this section.

Applications

The **Applications** tab shows all the running applications on your system. If you're having trouble getting an application to close normally, this is the place to go. To force an application to shut down, select the naughty application and click End Task, or right-click on the application and select End Task from the context menu. Be careful when using this feature! There is no "Are you sure?" prompt and it's easy to accidently close the wrong application.

There are two other handy buttons on the Applications tab:

- Switch To enables you to bring any program to the front (very handy when you have a large number of applications running).
- New Task is a lifesaver of a tool, enabling you to start any program you wish, as long as you know the filename of the program. One of the most common uses for New Task is to restart your desktop in Windows XP. If your desktop crashes, the Task Manager usually still works. Click on New Task and type in **explorer**. Sadly, this doesn't work in Windows Vista or Windows 7, but New Task is still a handy tool to start programs, especially if you don't have access to the Start menu.

Remember that everything is a process, so every application is also listed in the Processes tab. Right-click on an application and select Go To Process to open the Processes tab and see which process is running the application.

Processes

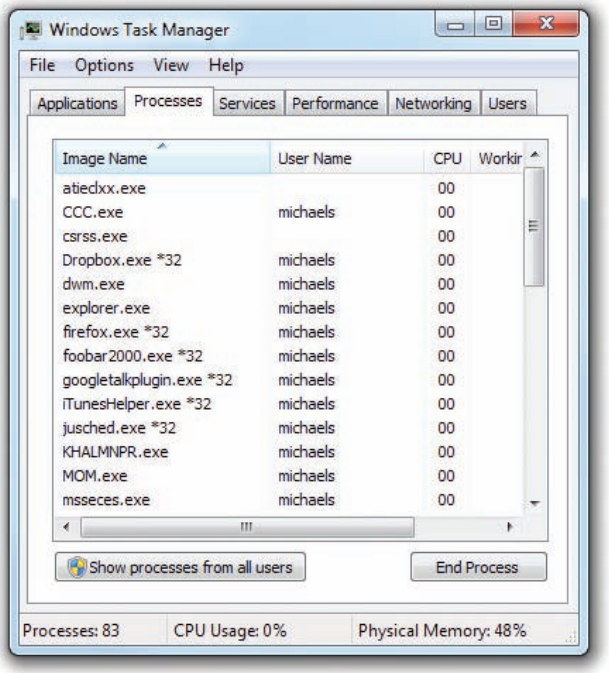
If you really want to tap the power of the Task Manager, you need to click on the **Processes** tab (see Figure 15.13). Since everything is a process, and the Processes tab shows you every running process, this is the one place

that enables you to see everything running on your computer.

All processes have certain common features that you should recognize:

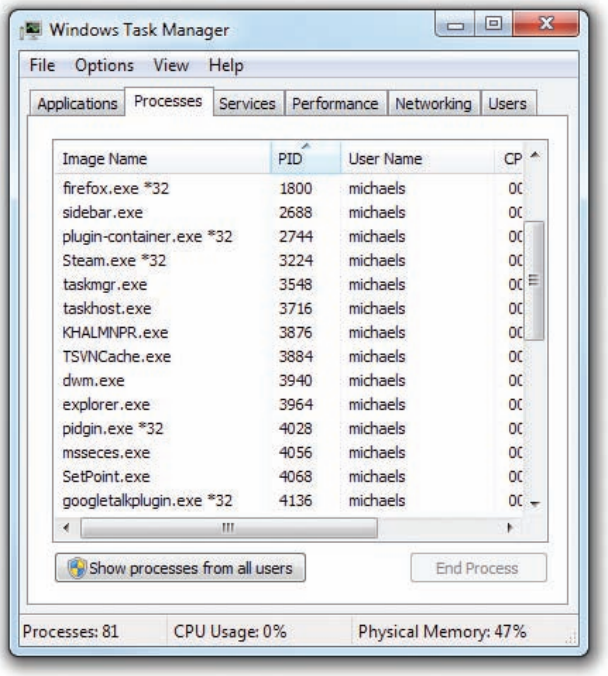
- A process is named after its executable file, which usually ends in .exe but can also end with other extensions.
- All processes have a user name to identify who started the process. A process started by Windows has the user name System.
- All processes have a Process Identifier (PID). To identify a process, you use the PID, not the process name. The Task Manager doesn't show the PID by default. Click on View | Select Columns and select the PID (Process Identifier) checkbox to see the PIDs (see Figure 15.14).

The Task Manager provides important information about processes. It shows the amount of CPU time (percentage) and the amount of RAM (in kilobytes) the process is using. Most processes also provide a description to help you understand what the process is doing, although you'll probably need to scroll right to see this information (see Figure 15.15).

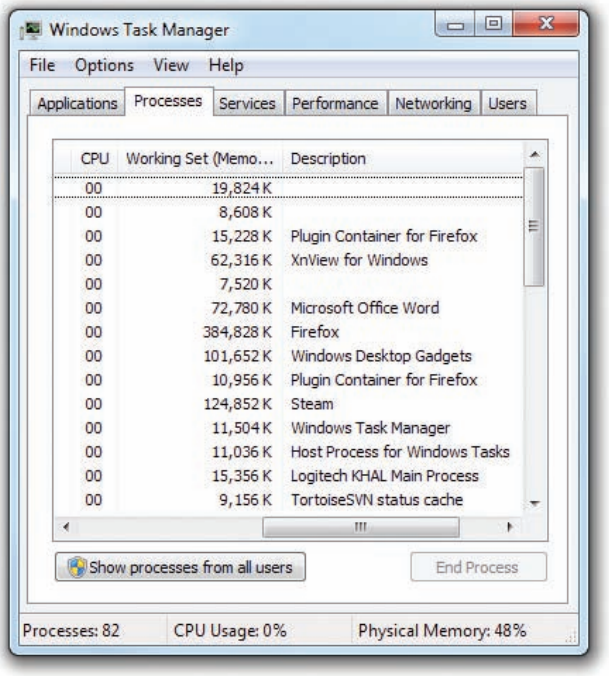


• **Figure 15.13** Processes tab in Windows 7

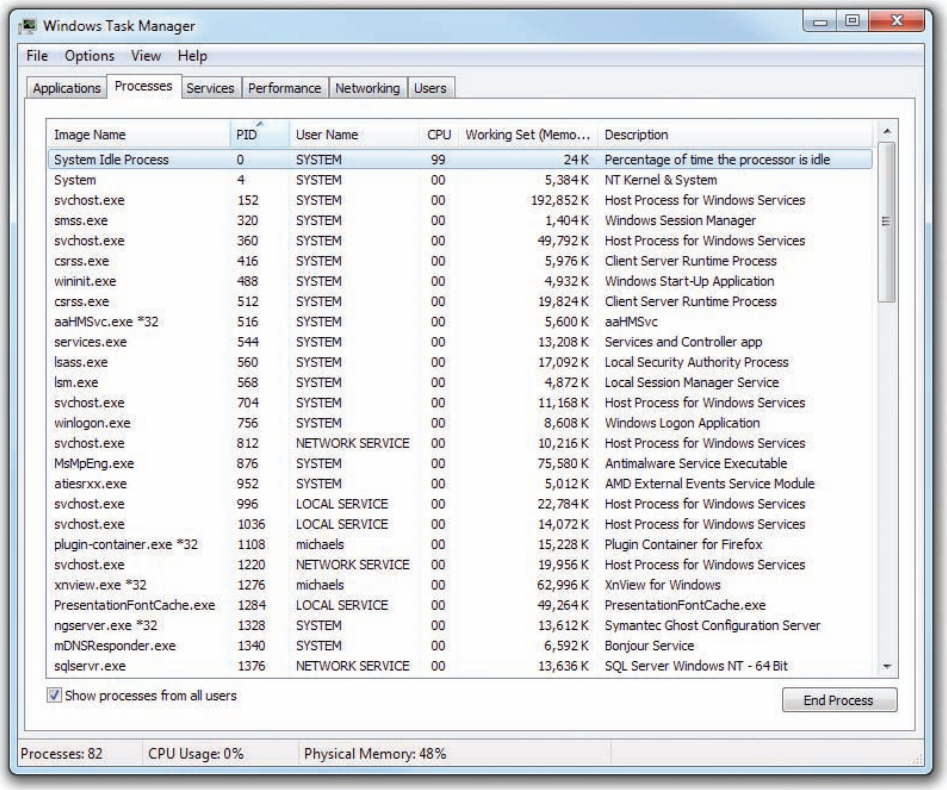
You'll notice that almost all of the processes have the same user name. By default, the Task Manager shows only processes associated with the current user. Click on *Show processes from all users* to see every process on



• **Figure 15.14** Processes tab showing the PID column in Windows 7



• **Figure 15.15** Processes details in Windows 7



• **Figure 15.16** Processes from all users in Windows 7

the system (see Figure 15.16). Note that some of the processes show a user name of Local Service or Network Service. As you might imagine, those are services!

Now that you understand the basics, let's watch the Task Manager do its magic with processes. If you select a process and click the End Process button, you'll instantly end that process. If the process is an application, that application will close.

Closing processes is important, but to take it even further, you need to select a process and right-click on it to see a number of options. If you select a process that's an application (the name of the process is a strong clue—winword.exe is Microsoft Word), you see something like Figure 15.17.

Open File Location takes you to wherever the file is located. This is extremely helpful when you're looking at a mysterious process and are trying to find out what it's doing on your computer.

You already know what End Process does. End Process Tree is extremely important but also complex, so let's save that for later.

Try This!

Closing Applications

Start up Notepad and then start up the Task Manager. Right-click on the Notepad application and select Go To Process. It takes you to the process. Right-click and select End Process to close the application.

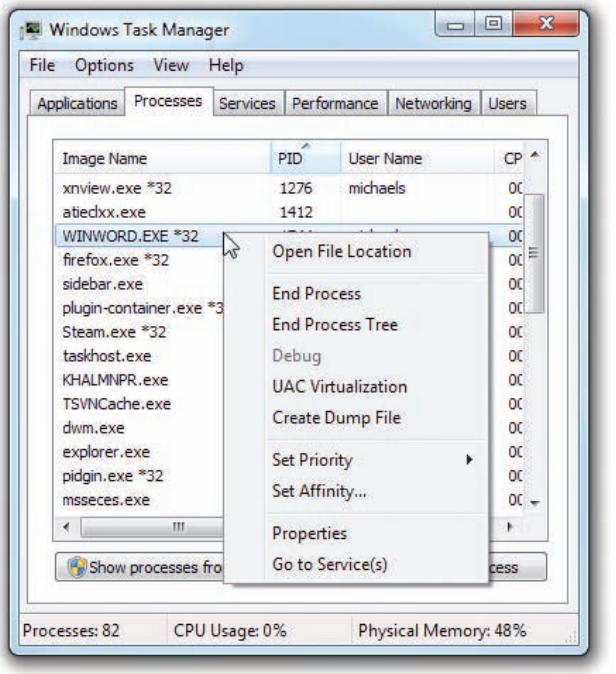
Debug is grayed out, unless you're running a Windows debugger program—see the explanation of dump files below.

UAC Virtualization gives older programs that weren't written to avoid accessing protected folders a way to do so by making a fake protected folder. In most cases, Windows handles this automatically, but there are rare cases where you'll need to set this manually. Again, you won't do this on your own—you'll be on the phone with the tech support for some software company and they'll tell you how to use UAC Virtualization.

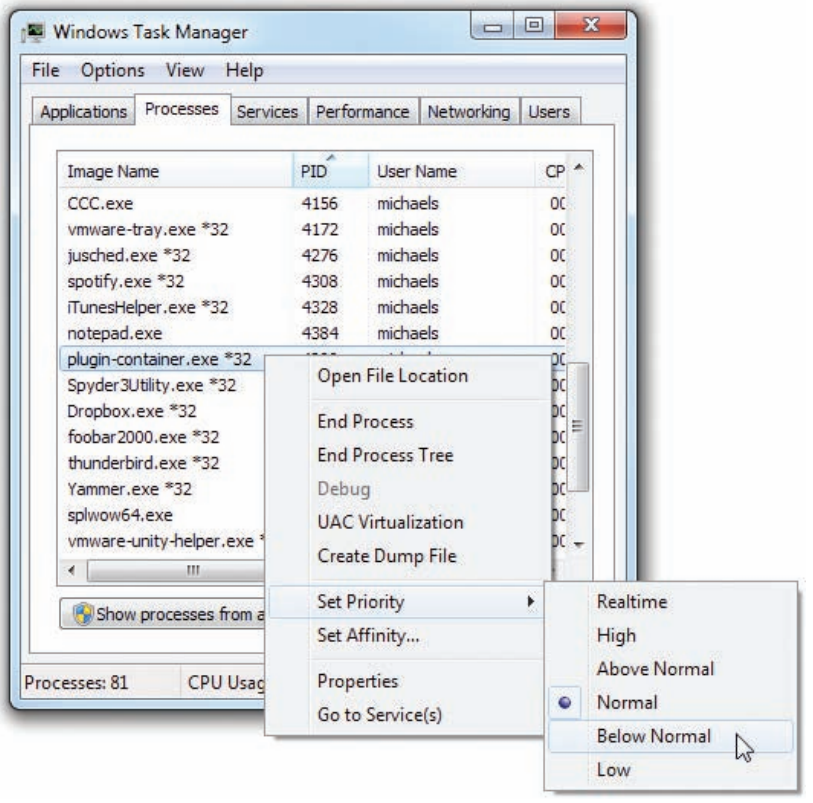
Dump files show the status of the program at the moment you click Create Dump File. Programmers use special debugging utilities to read dump files to analyze problems with programs. The only time you'd ever use this option is if you're having problems with a program and the support people ask you to make a dump file.

Set Priority gives you the ability to devote more or less processor time to a process (see Figure 15.18). This is very handy when you have a process that is slowing down your machine or if you have a process that is running too slowly and you want to speed it up.

Messing with priorities can get complicated quickly. The best idea here is to just increase the priority of a single

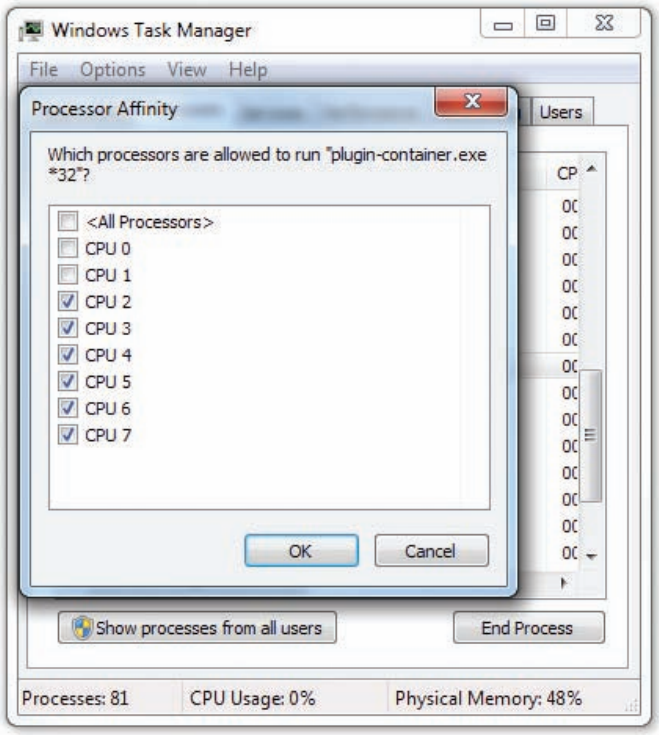


• **Figure 15.17** Processes detail of the right-click



• **Figure 15.18** Process priority

Setting any single process to Realtime priority will often bring the entire system to a crawl as no other process gets much CPU time—avoid Realtime priority.



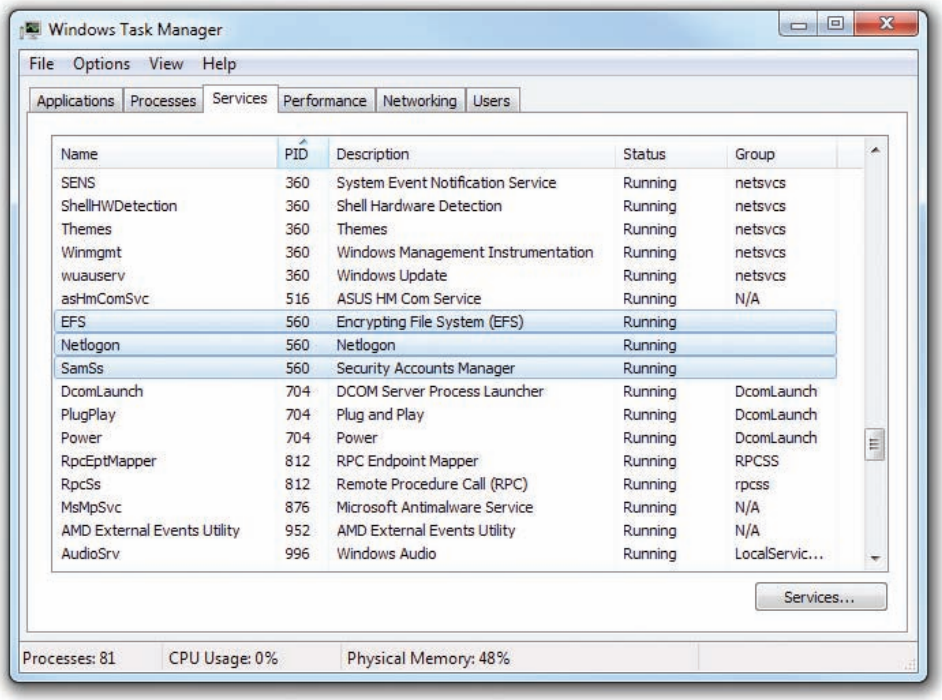
• **Figure 15.19** Turning off affinity to the first two cores

process you need to run faster or reduce the priority of a single process you need to run slower, without touching any other priorities.

Set Affinity is a handy way to give a process more CPU time without messing with priorities. Windows tends to use your first two CPUs more heavily than the others. If you have more than two cores on your CPU and you have a single process that uses a lot of CPU time, try unchecking the affinity checkbox for the first two cores to force the process to only use your other cores (see Figure 15.19).

The Properties option isn't too exciting. It's the same as if you were to right-click on the executable file and select Properties in Windows Explorer. Go to Service(s) will move you to the Services tab of the Task Manager, showing you any and all services associated with the process. Depending on the process, it could use no services or multiple services. This is a great tool for those "Program won't start because associated services aren't running" situations. Figure 15.20 shows what happens when you use Go to Service(s) for a process called lsass.exe.

Let's get back to the End Process Tree option. It's very common for a single process to be dependent on other processes (or for a process to start other processes). This creates a tree of dependencies. Sadly, the Task Manager doesn't give you any clue as to what processes depend on other processes,



• **Figure 15.20** Services associated with the lsass.exe process

but it still gives you the option to End Process Tree, which ends not only the process, but any process it depends on. At first glance, this is scary since it's very common for many processes to depend on one important process. Microsoft makes this less scary, as it will not let you kill a process tree for the most important system processes.

Even so, it would be nice to actually *see* what processes you're about to kill, wouldn't it? That's when the popular (and free) Process Explorer, written by Mark Russinovitch, is your go-to tool (see Figure 15.21)

Think of Process Explorer as the Task Manager on steroids. It's very powerful, and a lot of techs use it instead of the Task Manager. It isn't on the CompTIA A+ exams, but it should be. Instead of just listing all of the processes, Process Explorer uses a tree structure so you can see all the dependencies.

The Windows XP Task Manager lacks all of these context menu options except for End Process, End Process Tree, Debug, and Set Priority.

Process Explorer does so much more than just show a tree structure. Download a copy and play with it. You'll see why it's so popular.

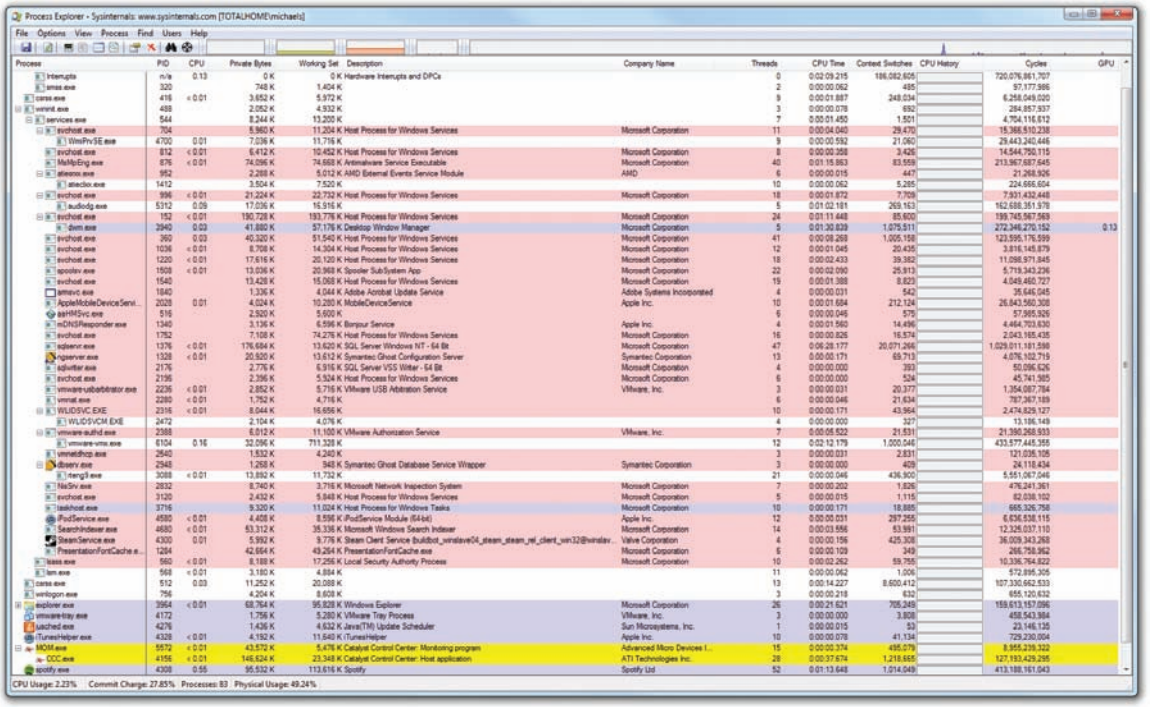
Services

If you've got Windows Vista or Windows 7, you can use the **Services** tab in the Task Manager to work with services directly (see Figure 15.22). Here, services can be stopped or started, and you can go to the associated process.

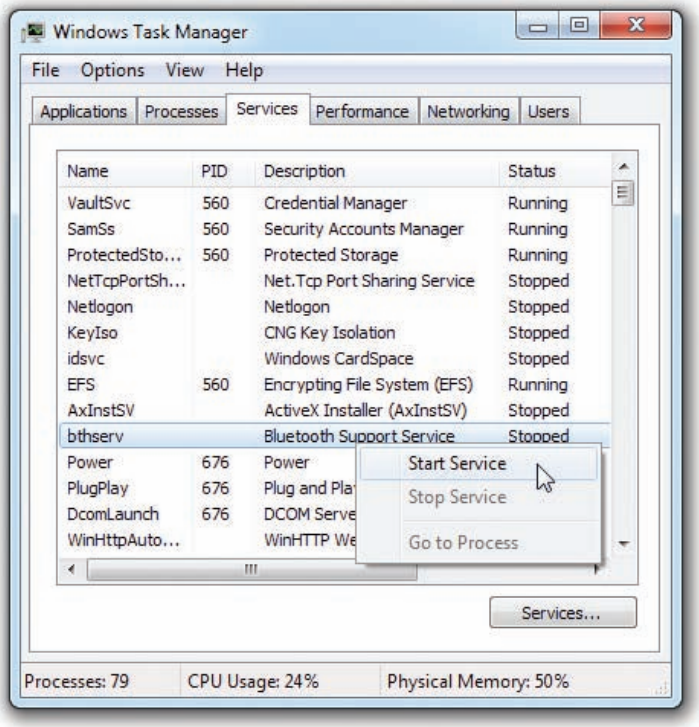
The best way to see services in action is to use the Services Control Panel applet. To open it, click on the Services button at the bottom of the Services tab in the Task Manager or go to Control Panel | Administrative Tools | Services. Figure 15.23 shows the Services applet running in Windows 7.

Look closely at Figure 15.23. Each line in this applet is an individual service. Services don't have their own window, so you use the Services applet to start, stop, and configure them. You can see if a service is running by reading the Status column. To configure a service, right-click on the service name. The context menu enables you to start, stop, pause, resume, or restart

You can open the Services applet from the Start | Run dialog box or Start | Search bar. Type **services.msc** and press ENTER.



• **Figure 15.21** Process Explorer



• **Figure 15.22** Services tab in Task Manager

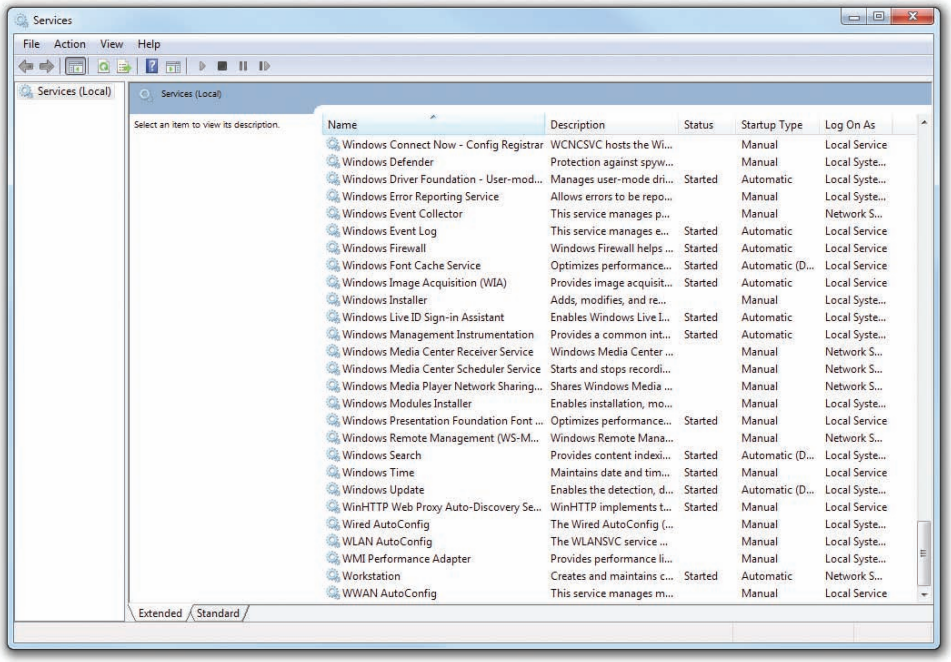
any service. Click on Properties to see a dialog box similar to the one shown in Figure 15.24.

Of the four tabs you see in the Properties dialog box, General and Recovery are by far the most used. The General tab provides the name of the service, describes the service, and enables you to stop, start, pause, or resume the service. You can also define how the service starts: Manual (you go here to start it), Automatic (starts at beginning of Windows boot), Disabled (prevents the service from starting in any fashion), or Automatic (delayed start), which starts the service at boot but only after pretty much everything else has started.

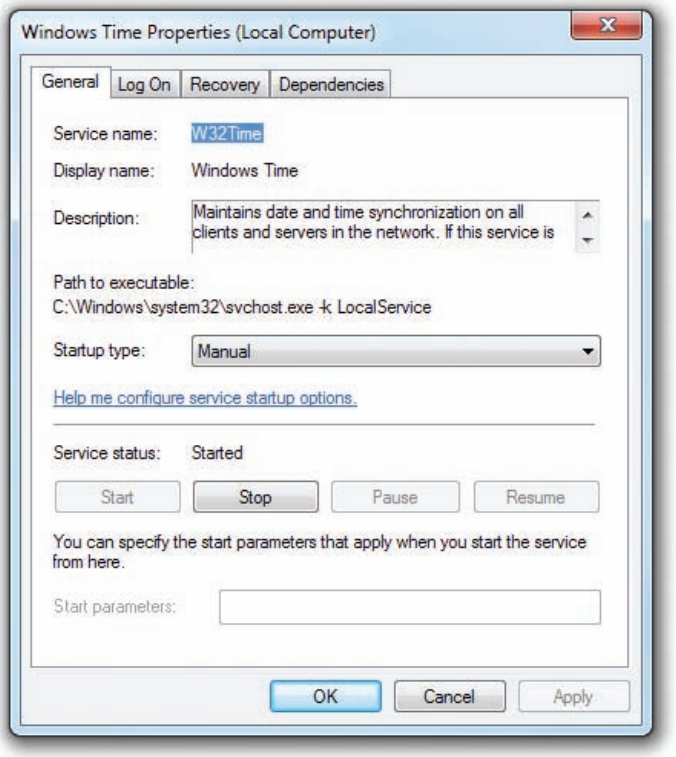
Performance

For optimization purposes, the Task Manager is a great tool for investigating how hard your RAM and CPU are working at any given moment and why. Click the **Performance** tab to reveal a handy screen with the most commonly used information: CPU usage, available physical memory, size of the disk cache, commit charge (memory for programs), and kernel memory (memory used by Windows). Figure 15.25 shows a system with an eight-core processor, which is why you see eight graphs under CPU Usage History. A system with a single-core processor would have a single screen.

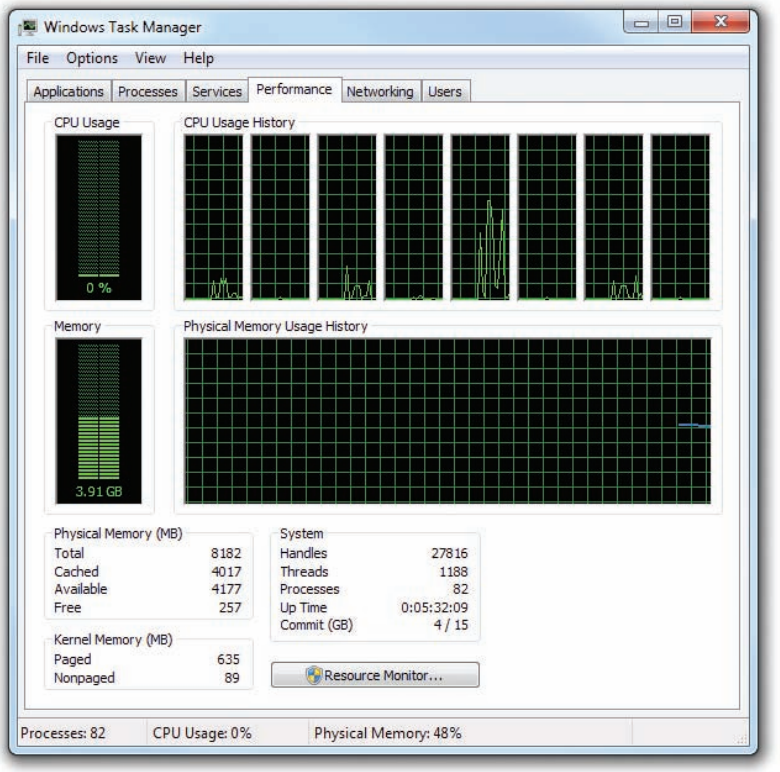
You can start or stop any service at a command prompt by typing **net start** (or **stop**) *service name*.



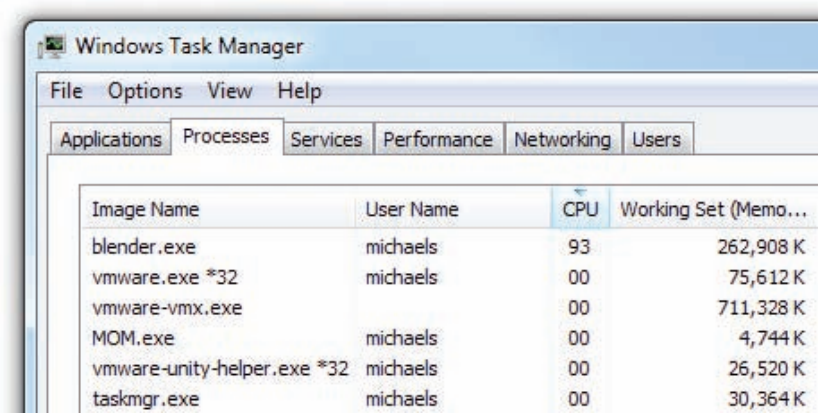
• **Figure 15.23** Services applet



• **Figure 15.24** Service Properties dialog box



• **Figure 15.25** Task Manager Performance tab



• **Figure 15.26** CPU usage

Not only does the Task Manager tell you how much CPU and RAM usage is taking place, it also tells you what program is using those resources. Let's say your system is running slowly. You open the Task Manager and see that your CPU usage is at 100 percent. You then click on the Processes tab to see all the processes running on your system. Click on the CPU column heading to sort all processes by CPU usage to see who's hogging the CPU (see Figure 15.26). If necessary, shut down the program or change its priority to fix the issue.

Networking and Users

The other two tabs in the Task Manager, Networking and Users, enable you to see network use at a glance and see which users' accounts are currently logged on to the local machine. The Networking tab is a good first spot to look if you think the computer is running slowly on the network. If there's little activity displayed, then it's not traffic from your computer that's causing the slowdown, so you need to look elsewhere. Chapter 22 covers network troubleshooting in a lot more detail, so we'll leave the Networking tab alone for now.

The Users tab enables you to log off other users if you have the proper permissions. You can also log off from here. There's not much else to say, but since the CompTIA A+ 220-802 exam objectives want this tab used in a scenario, here's one hypothetical. Another user is still logged on and left a critical file open that you need to access. Logging the user off forces his or her applications to close and makes the file available. Of course, unsaved changes will be lost, so use caution here.

The tasklist and taskkill Commands

The two command-line utilities **tasklist** and taskkill enable you to work with tasks, similarly to what you can do with the Task Manager. The tasklist command enables you to view running processes on a local or remote system. Open up a command prompt and type **tasklist**. The following is a partial example of the output:

```
C:\Users\mike>tasklist

Image Name                     PID Session Name        Session#    Mem Usage
========================= ====== ================ ========== ============
System Idle Process              0 Services                   0         24 K
System                           4 Services                   0        940 K
smss.exe                       268 Services                   0        340 K
csrss.exe                      372 Services                   0      2,388 K
wininit.exe                    444 Services                   0        968 K
csrss.exe                      452 Console                    1      9,788 K
winlogon.exe                   500 Console                    1      2,420 K
```

The CompTIA A+ 802 exam objectives mention a command-line utility called *tlist*. Tasklist replaced tlist back in Windows XP, but for some reason CompTIA included it anyway.

```
services.exe              544 Services                   0      4,536 K
svchost.exe               756 Services                   0      4,320 K
MsMpEng.exe               828 Services                   0     42,164 K
atiesrxx.exe              904 Services                   0        824 K
notepad.exe              3932 Console                    0      6,612 K
```

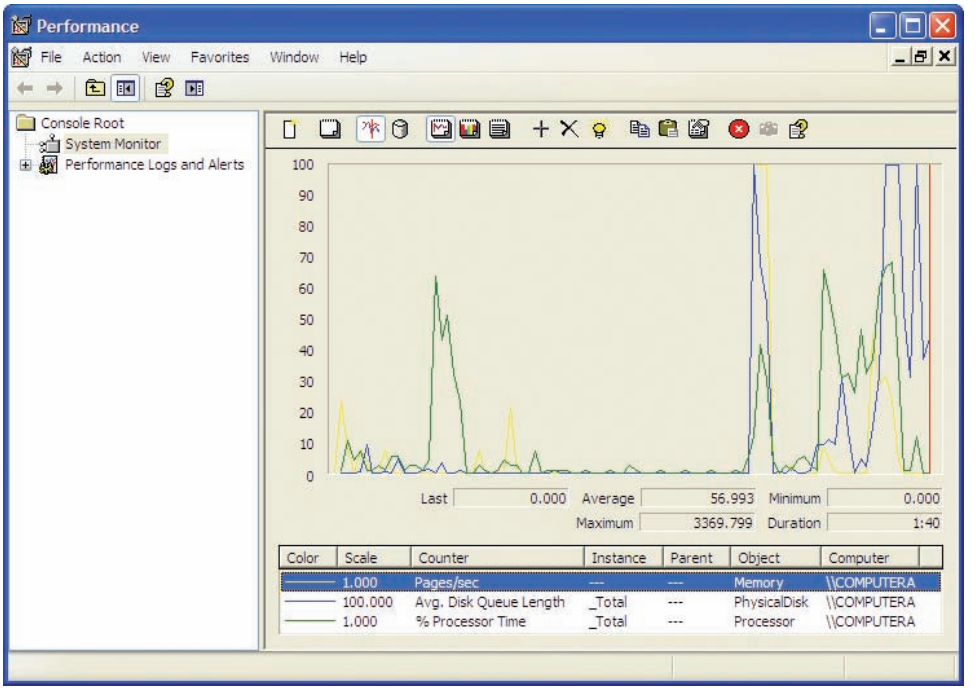
You can kill a process using the taskkill command. See the notepad.exe listing in the preceding tasklist output? You can kill the process using either the name or the PID, as shown here:

Performance Console

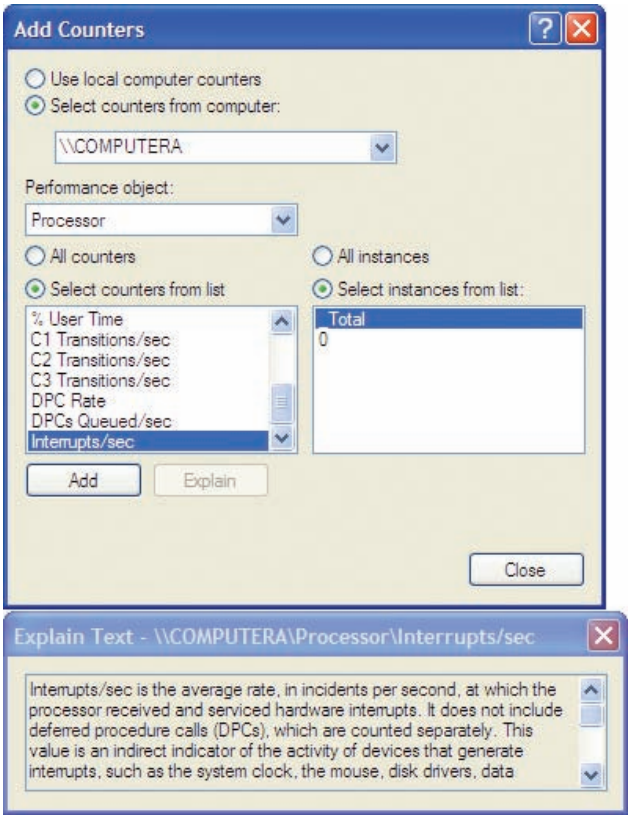
The Task Manager is good for identifying current problems, but what about problems that happen when you're not around? What if your system is always running at a CPU utilization of 20 percent—is that good or bad? Windows XP provides a tool called the **Performance console** that logs resource usage so you can track items such as CPU and RAM usage over time. Performance is an MMC console file, perfmon.msc, so you call it from Start | Run or through the Performance icon in Administrative Tools. Use either method to open the Performance console (see Figure 15.27). As you can see, there are two nodes, System Monitor and Performance Logs and Alerts.

Objects and Counters

To begin working with the Performance console, you need to understand two terms: object and counter. An **object** is a system component that is given



• Figure 15.27 Performance console



• **Figure 15.28** Add Counters dialog box

a set of characteristics and can be managed by the operating system as a single entity. A **counter** tracks specific information about an object. For example, the Processor object has a counter, %Processor Time, that tracks the percentage of elapsed time the processor uses to execute a non-idle thread. Many counters can be associated with an object.

System Monitor

System Monitor gathers real-time data on objects such as memory, physical disk, processor, and network, and displays this data as a graph (line graph), histogram (bar graph), or simple report. When you first open the Performance console, System Monitor shows data in graph form. The data displayed is from the set of three counters listed below the chart. If you want to add counters, click the Add button (the one that looks like a plus sign) or press CTRL-I to open the Add Counters dialog box. Click the Performance object drop-down list and select one of the many different objects you can monitor. The Add Counters dialog box includes a helpful feature: you can select a counter and click the Explain button to learn about the counter, as shown in Figure 15.28. Try that now.

Even with just three counters selected, the graph can get a little busy. That's where one of my favorite System Monitor features shines. If you want the line of charted data from just one counter to stand out, select the counter in the list below the graph and then press CTRL-H. See how this trick makes the %Processor Time line stand out in Figure 15.29? Imagine how useful that is when you are monitoring a dozen counters.

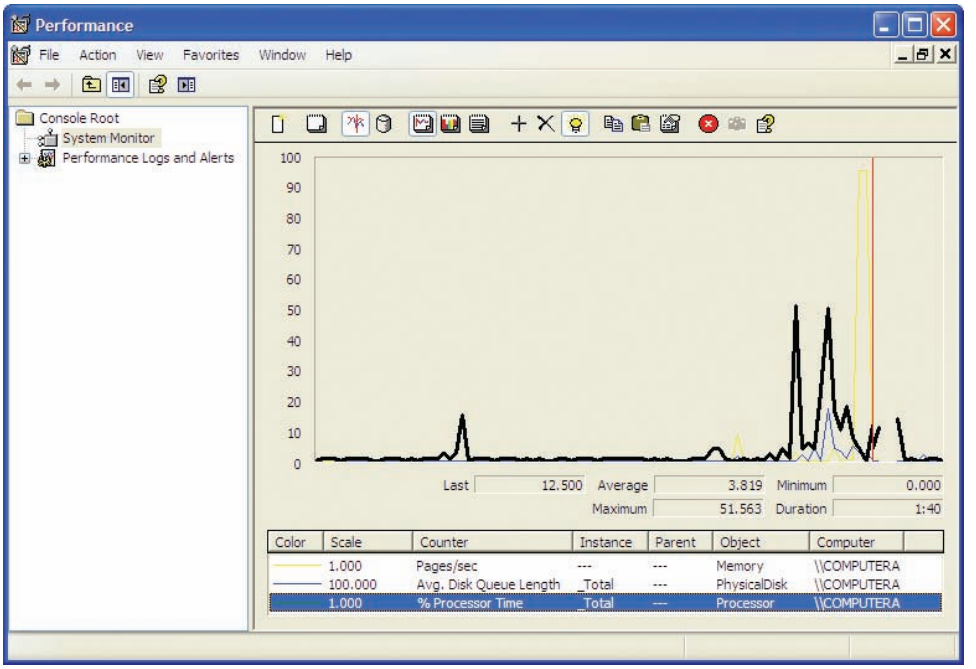
Performance Logs and Alerts

The **Performance Logs and Alerts** snap-in enables Windows to create a written record of just about anything that happens on your system. Do you want to know if someone is trying to log on to your system when you're not around?

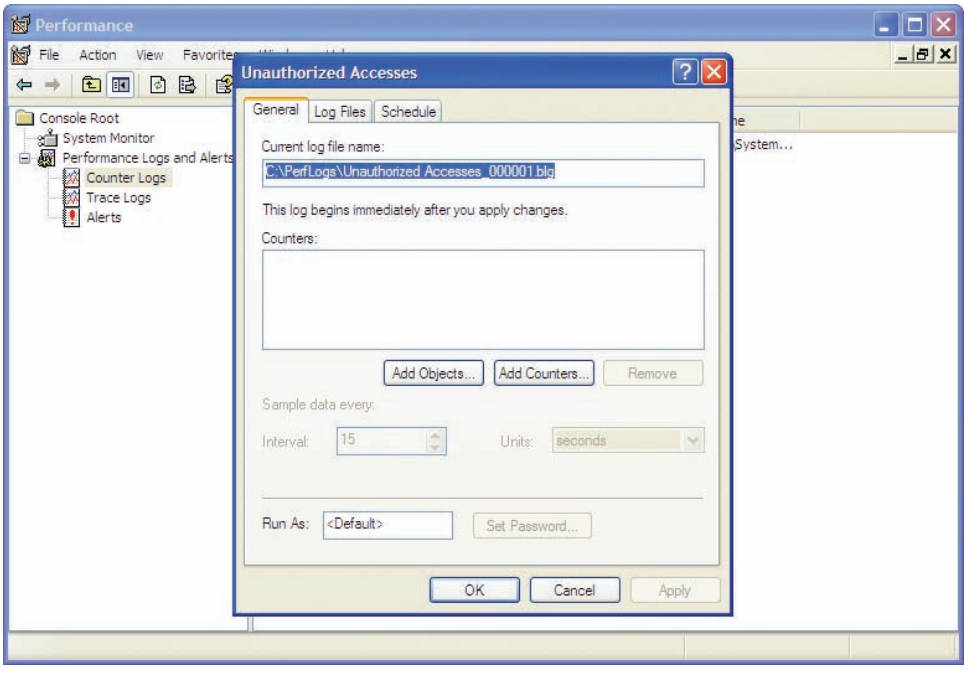
To create the new event log, right-click Counter Logs and select New Log Settings. Give the new log a name—in this example, "Unauthorized Accesses." Click OK, and a properties dialog box for the new log appears, similar to that shown in Figure 15.30.

To select counters for the log, click Add Counters and then select the *Use local computer counters* radio button. Select Server from the Performance object pull-down menu and then select Errors Logon from the list of counters; click Add and then click Close.

Back in the properties dialog box for your new log, click the Schedule tab and set up when you want this thing to start running—probably at the end of the workday today. Then select when it should stop logging—probably tomorrow morning when you start work. Click the Log Files tab to see where the log file will be saved—probably C:\PerfLogs—and make a



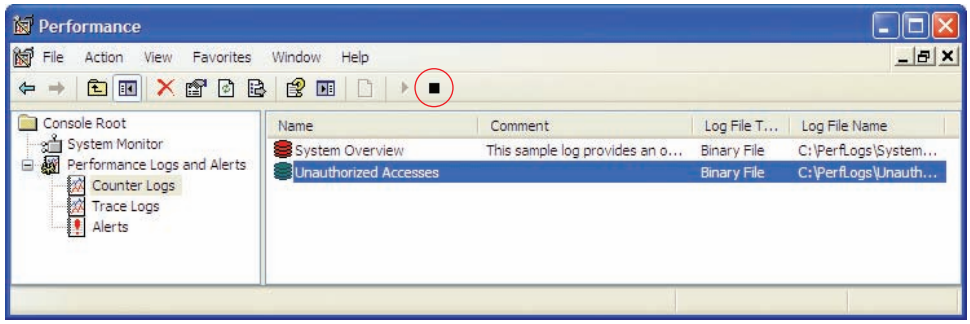
• **Figure 15.29** Pressing CTRL-H makes one set of data stand out.



• **Figure 15.30** Creating a new performance log

note of the filename. The filename will consist of the name you gave the log and a number. In this example I named the new performance log "Unauthorized Accesses," so the filename is Unauthorized Accesses_000001.blg.

When you come back in the morning, open the Performance console, select Performance Logs and Alerts, and then select Counter Logs. Your



• **Figure 15.31** Stopping the performance log

log should be listed on the right. The icon by the log name will be green if the log is still running or red if it has stopped. If it has not stopped, select it and click the Stop button (the one with the black square, circled in Figure 15.31).

To view the log, open the Performance console, select System Monitor, change to Report view, and load the file as a new source by using the log's properties dialog box.

Performance Tools in Windows Vista and Windows 7

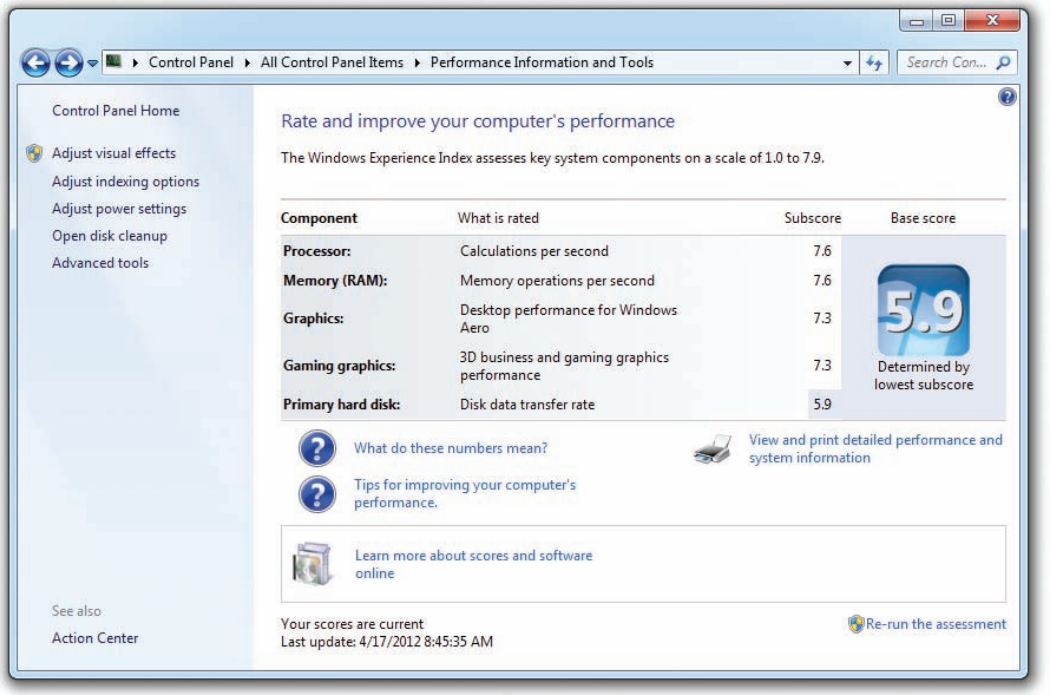
Microsoft improved on the Performance console dramatically with Reliability and Performance Monitor in Windows Vista and yet again with Performance Monitor in Windows 7. **Reliability and Performance Monitor** still has a complete Performance console with all the objects and counters you see in Windows XP, but it adds an excellent Resource Overview, a Reliability Monitor tool, and a much more flexible way to use counters with Data Collector Sets and Reports. **Performance Monitor** in Windows 7 functions almost identically to Reliability and Performance Monitor in Windows Vista, but Microsoft pulled out the Reliability tool to make the remaining tool smaller and tighter.

The Windows Experience Index runs on a scale from 1.0 to 7.9 and gives you a pretty good gauge of the relative strengths and weaknesses of a system. The system in Figure 15.32, for example, has a CPU near the top of the chart (it's an Intel Core i7) and lots of fast RAM (8 GB), but a relatively modest hard drive. So the next obvious upgrade to this system would be to move from a platter-based drive to a solid-state drive. Disk performance would certainly jump dramatically.

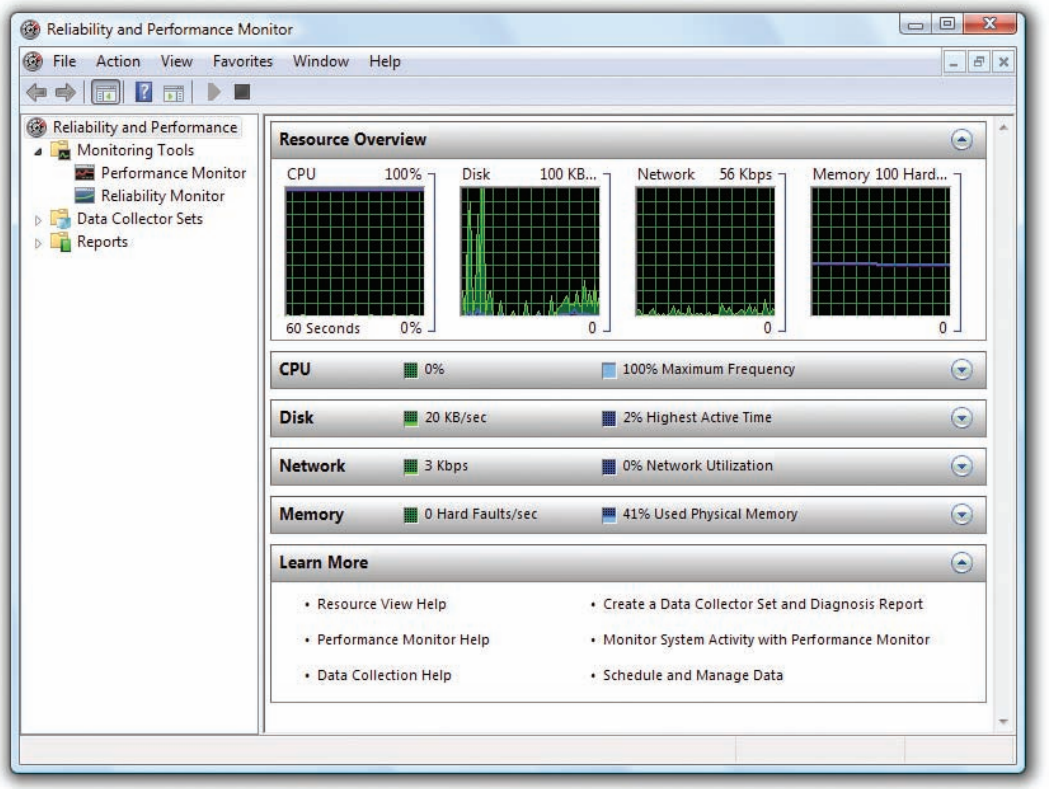
You can open Reliability and Performance Monitor/Performance Monitor by starting the Performance Information and Tools applet in Control Panel. The initial screen is the *Rate and improve your computer's performance* screen (see Figure 15.32), which shows how your computer's hardware stacks up in the Windows Experience Index. Click the Advanced tools link in the left task area. Then click on Open Reliability and Performance Monitor or Open Performance Monitor, depending on the Windows version. You can also open the tool by going to Start | Search bar, typing **perfmon.msc**, and pressing ENTER.

Reliability and Performance Monitor in Windows Vista opens to a Resource Overview screen (see Figure 15.33). Think of the Resource Overview as an advanced Task Manager, giving details on CPU, hard drive, network, and memory usage.

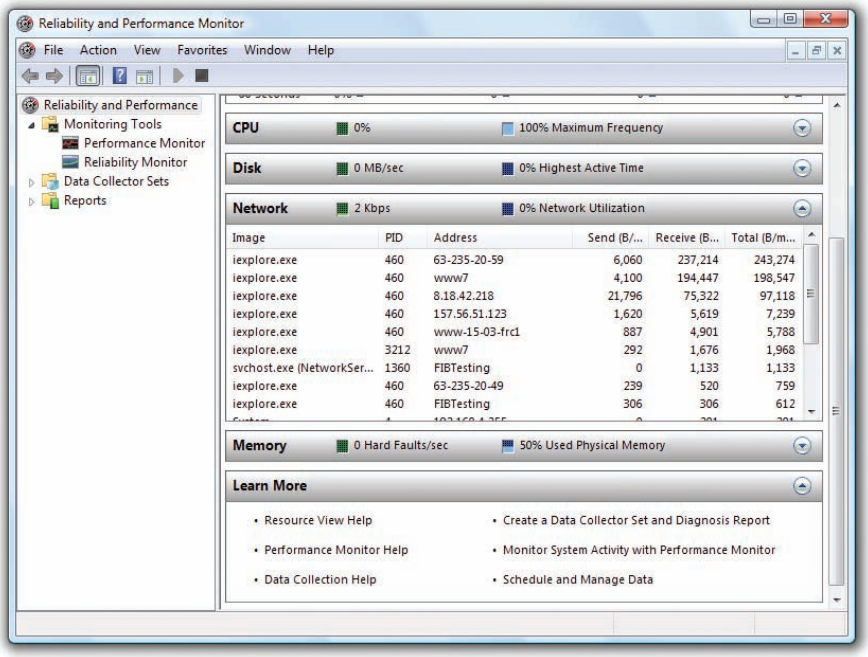
When you click on one of the four bars, you get details on exactly which processes are using those resources—a powerful tool when you suspect a



• **Figure 15.32** Performance Information and Tools showing the computer's rating on the Windows Experience Index



• **Figure 15.33** Resource Overview in Windows Vista

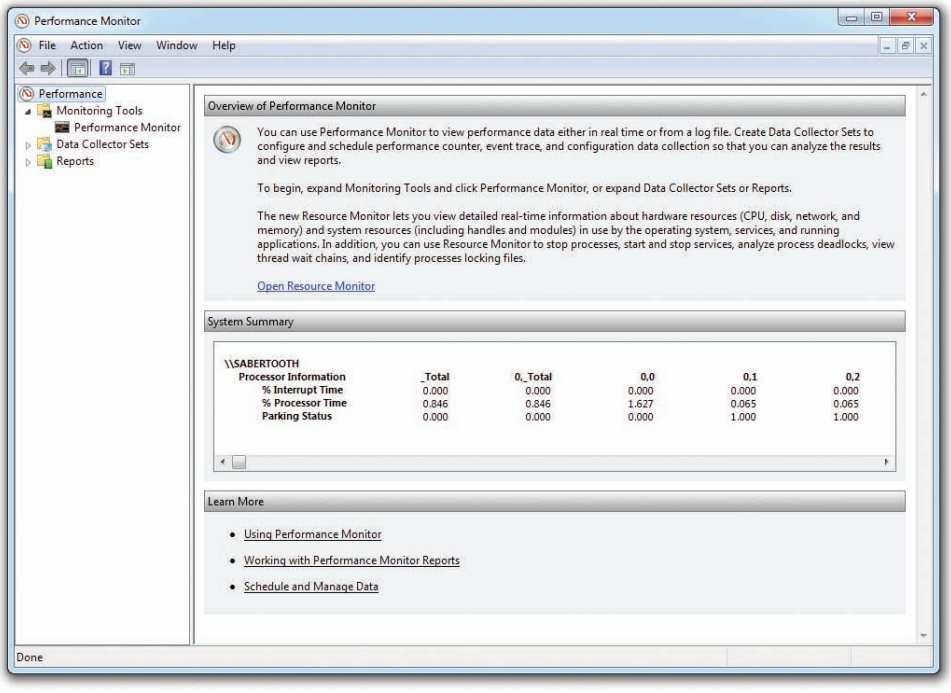


• **Figure 15.34** Network bar in Resource Overview

The Reliability Monitor tool gives you an overview of how a PC has behaved over time, showing important events such as application or OS crashes. You can find the tool in Windows 7 as part of the Action Center Control Panel applet.

program might be hogging something! Figure 15.34 shows the Network bar opened to reveal the processes using the network and how much data each is sending.

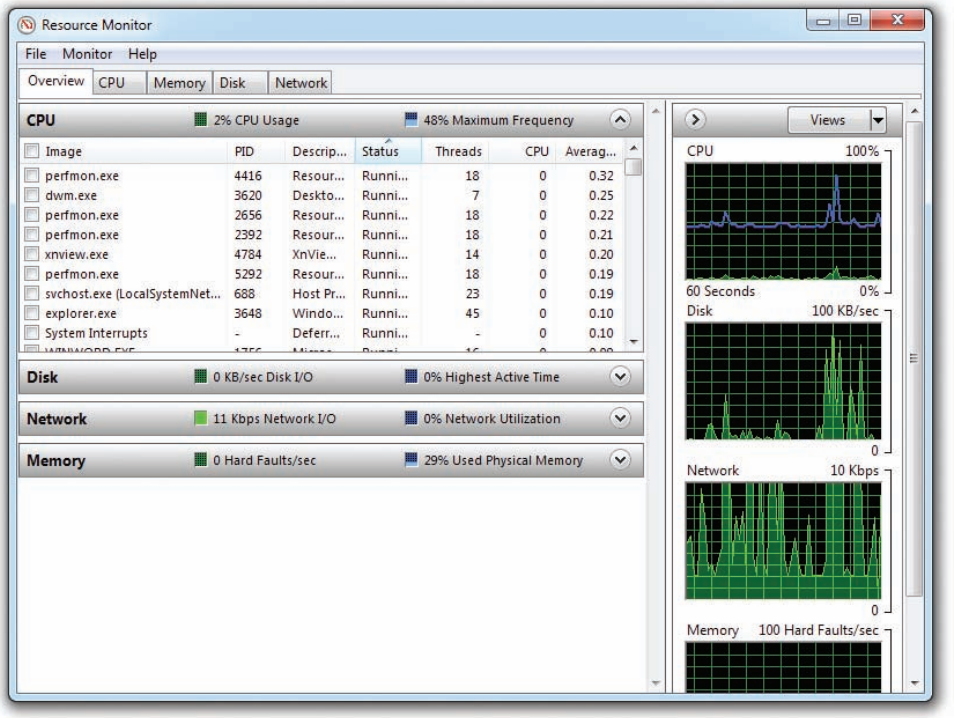
Performance Monitor in Windows 7 opens to a more modest screen that displays some text about Performance Monitor and a System Summary (see



• **Figure 15.35** Initial Performance Monitor screen in Windows 7

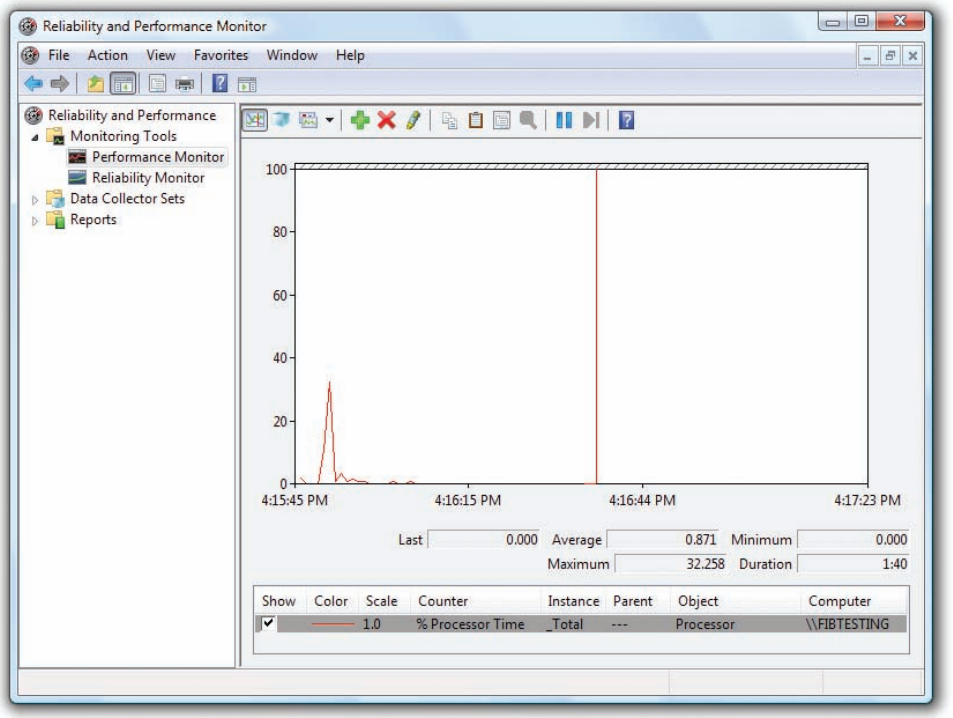
Figure 15.35). You can get to the Overview screen by clicking the Open Resource Monitor link on the main screen. Aside from orienting the graphical screens on the right rather than on the top, the tool is the same as the Resource Overview in Windows Vista (see Figure 15.36).

The Performance Monitor option you can select under Monitoring Tools in either version of the OS is simply a re-creation of the Performance console and works as described earlier for Windows XP (see Figure 15.37). This is a great tool for quick checks on specific counters.

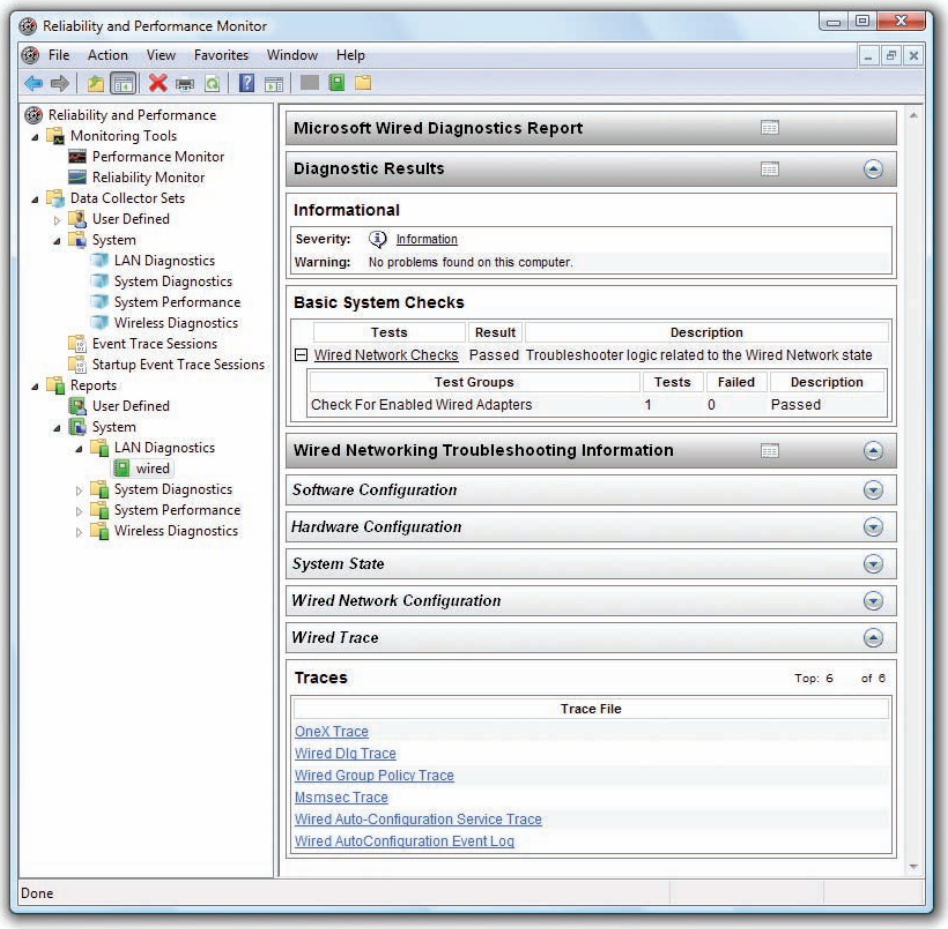


• **Figure 15.36** Resource Monitor displaying CPU usage

Microsoft included Data Collector Sets in Reliability and Performance Monitor and Performance Monitor, groupings of counters you can use to make reports. You can make your own Data Collector Sets (User Defined) or you



• **Figure 15.37** Reliability and Performance Monitor



• **Figure 15.38** Sample report

can just grab one of the predefined system sets. Once you start a Data Collector Set, you can use the Reports option to see the results (see Figure 15.38). Data Collector Sets not only enable you to choose counter objects to track, but also enable you to schedule when you want them to run.

The CompTIA A+ exams aren't going to ask too many detailed questions on either Performance Monitor or Reliability and Performance Monitor. That doesn't mean you can ignore these amazing tools! Make sure you understand that these tools give you the power to inspect anything happening on your system to help you diagnose problems.

Tools for Programmers

Back in Chapter 4, I discussed many of the tools available under Administrative Tools in Control Panel. I left out two applets that I want to discuss now for two reasons: first, because they're covered on the CompTIA A+ 220-802 exam, and second, because they deal with some low-level functionality in Windows that affects how a lot of applications are programmed. Read on to find out more about the Component Services and Data Sources (ODBC) applets.

Component Services

To understand all that **Component Services** can do would require a huge amount of information—far greater than the scope of CompTIA's A+ exams.

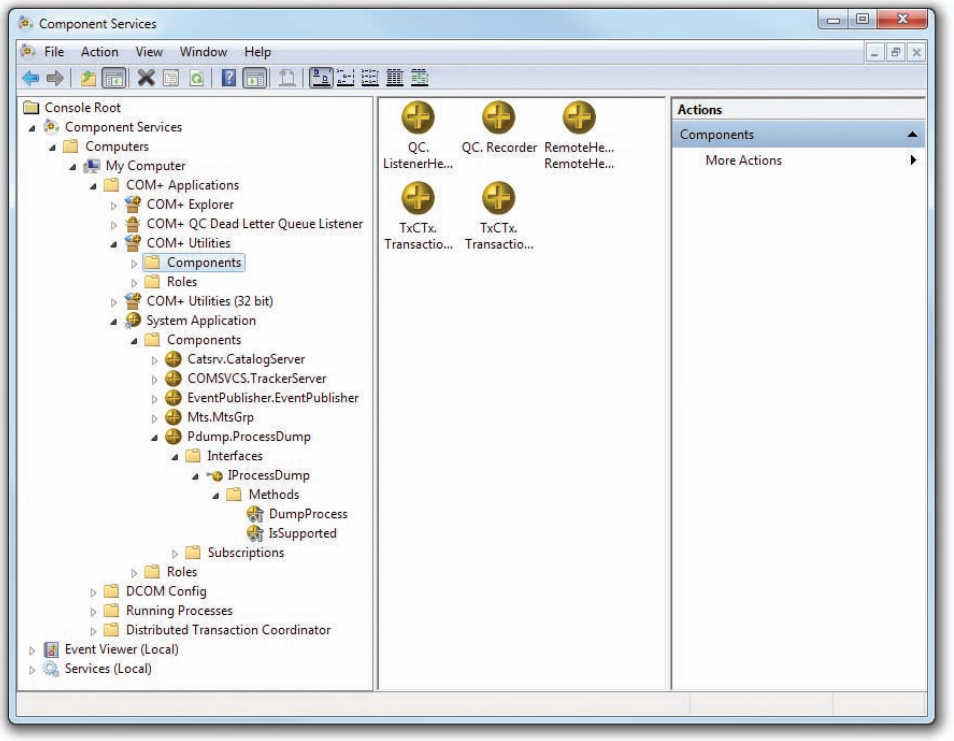
Simply put, as long as Windows has existed, Microsoft has come up with many tools (with names like COM, DCOM, and COM+) to enable programmers to share data objects (an element of programs) between applications on a single computer. Over time, this sharing was extended so that you could share objects between computers on a network.

In almost all cases, this object sharing doesn't require you to do anything more than install an application that uses these features. Component Services is there, however, for those very rare times when something's either wrong or a programmer needs you to make manual changes (see Figure 15.39). If you have a company that creates in-house or buys custom applications, there's a better than good chance that you'll be firing up Component Services and working with programmers, manually installing programs, and tweaking those programs to get them to work the way you wish. Professional, third-party applications (the kind you buy in stores) should automatically configure any of these programs during the installation process, making it extremely rare that you'll need to go into Component Services.

Every version of Windows has Component Services, but there's no ready-to-go icon in Windows XP or Windows Vista. You'll need to make a custom MMC and load Component Services from there.

Data Sources

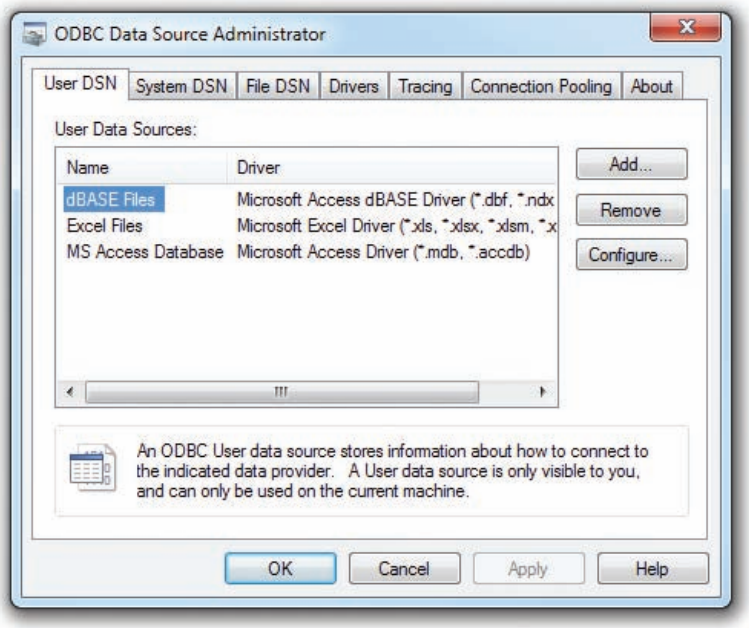
One of the oldest and most common motivations to make networks is the idea of a number of computers accessing one or more shared databases. These computers might not all be running the same operating system, nor



• **Figure 15.39** Component Services in Windows 7

will they always use the same application to access those databases. That's where Open Database Connectivity (ODBC) really shines. ODBC is a coding standard that enables programmers to write databases and the applications that use them in a way that they can query ODBC to see how to locate and access a database without any concern about what application or operating system is used.

Microsoft's tool to configure ODBC is called **ODBC Data Source Administrator** (see Figure 15.40). Data Source Administrator enables you to create and manage entries called Data Source Names (DSNs) that point ODBC to a database. DSNs are used by ODBC-aware applications to query ODBC to find their databases. Keep in mind that you'll rarely go into Data Source Administrator unless you're making your own shared databases.



• **Figure 15.40** Data Source Administrator in Windows 7

Chapter 15 Review

Chapter Summary

After reading this chapter and completing the exercises, you should understand the following about Windows under the hood.

Work with the Registry

- The Registry is a huge database that stores everything about your PC, including information on all of the hardware in the PC, network information, user preferences, file types, and more. Registry files are stored in the \%SystemRoot%\System32\config folder.
- To edit the Registry, you use regedt32 (or regedit), also known as the Registry Editor.
- The Registry is organized in a tree structure similar to the folders on a PC. The five root keys of the Registry are HKEY_CLASSES_ROOT, HKEY_CURRENT_USER, HKEY_USERS, HKEY_LOCAL_MACHINE, and HKEY_CURRENT_CONFIG.
- Root keys are divided into subkeys, which are divided into values. In the Registry Editor, keys are listed on the left and values are listed on the right.
- The Registry Editor's export feature enables you to save either the full registry or only a single rootkey or subkey (with all subkeys and values under it).

Understand and observe in detail the Windows boot process

- The Windows installation creates several files and folders that an OS needs to run a PC.
- Each version of Windows boots differently. Windows XP was the last version of Windows to use the ntldr boot process. Windows Vista and Windows 7 boot using Windows Boot Manager (bootmgr).
- Windows XP distinguishes between the files that start the OS (system files) and the rest of the OS files. The system files include ntldr (NT Loader), boot.ini, and ntdetect.com. If you boot from a SCSI hard drive, you also need ntbootdd.sys.
- To boot Windows XP, the CPU wakes up and runs the system BIOS. BIOS looks for the first bootable drive's partition table and finds a boot sector marked as active. The boot sector takes control from BIOS and reads the master file table (MFT). The MFT points to the Windows XP system files. The system files point to the boot files (which include ntoskrnl.exe, the \Winnt\System32\Config\System file, and the device drivers) and the GUI loads. The OS is then up and running and ready to work.
- The Windows XP OS files can reside on any partition or volume in the PC.
- Windows Vista and Windows 7 use a very different boot process than Windows XP. Windows Vista/7 supports both BIOS and UEFI; the process is slightly different between the two.
- In a BIOS-based system, BIOS loads, and looks for the master boot record (MBR). The MBR scans for a system partition and loads the boot sector. The boot sector points to the Boot Manager (bootmgr) and Windows loads.
- In a UEFI system, neither the MBR/GPT nor the file system boot code is run. UEFI simply loads Boot Manager directly.
- In addition to booting Windows, Boot Manager is also responsible for controlling dual-boot systems. The Boot Configuration Data (BCD) file contains information about the various OSs installed on the system, along with instructions for how to load each of them.

Control processes and services

- In Windows, programs are executable files waiting on a mass storage device. When you start a program, Windows loads it into RAM as a process. The CPU reads the process and the process tells the CPU which chunks of code to run.
- The Windows Task Manager is the one-stop-shop for anything to do with applications, processes, and services (Windows Vista/7). To open the Task Manager, press CTRL-SHIFT-ESC or CTRL-ALT-DELETE and select Task Manager. Alternatively, go to Start | Run or Start | Search, type **taskmgr**, and press ENTER.

- The Applications tab shows all the running applications on your system. Here, you can force an unruly application to shut down. You can also switch to open applications, start a new task (open an application), or see the appropriate process for the application.
- The Processes tab lists all the running processes, including any running applications and services. The list of processes shows the name and the user who started the process. You can also display the Process Identifier (PID), which is how Windows refers to each process.
- From the Processes tab, you can end a process, find the executable associated with the process, debug, set UAC Virtualization, create a dump file, set priority and affinity, and see the associated services.
- The Task Manager will show you all the services on your computer, but for more options, you should use the Services applet in Administrative Tools. Here you can start, stop, and configure all of your services. You can also set if and when the service will load when you boot Windows.
- The Performance tab shows you CPU usage, memory usage, and the size of the disk cache. You can switch to the Processes tab to see which process is using your resources.
- The Users tab enables you to log off your own account or another user's account.
- The tasklist command is the command-line version of the Task Manager. Here, you can view and end processes by name or PID.
- The Performance console enables you to create logs that track CPU, memory, and other resource usages over time. To track these resources, you must set up objects (the resources you wish to track) and counters (the measurements used to track the objects).
- System Monitor gathers real-time data on objects such as memory, physical disk, processor, and network, and displays this data as a graph (line graph), histogram (bar graph), or simple report. Think of System Monitor as a more detailed, customizable Task Manager.
- The Performance Logs and Alerts snap-in enables Windows to create a written record of just about anything that happens on your system.
- Reliability and Performance Monitor in Windows Vista and Performance Monitor in Windows 7 improve on Windows XP's Performance console. They include the older console, but add the Resource Overview/Resource Monitor, Reliability Monitor (Vista only), and Data Collector Sets and Reports.
- Think of the Resource Overview/Resource Monitor as an advanced Task Manager, giving details on CPU, hard drive, network, and memory usage. When you click on one of the four bars, you get details on exactly which processes are using those resources—a powerful tool when you suspect a program might be hogging something.
- Microsoft included Data Collector Sets in Reliability and Performance Monitor/Performance Monitor, groupings of counters you can use to make reports. You can make your own Data Collector Sets (User Defined) or you can just grab one of the predefined system sets.

Explore Windows tools for programmers

- Component Services enables programmers to share data objects between applications on a single computer or over a network. Professional, third-party applications (the kind you buy in stores) should automatically configure any of these programs during the installation process, making it extremely rare that you'll need to go into Component Services.
- ODBC is a coding standard that enables programmers to write databases and the applications that use them in a way that they can query ODBC to see how to locate and access a database without any concern about what application or operating system is used. Microsoft's tool to configure ODBC is called Data Source Administrator. Data Source Administrator enables you to create and manage entries called Data Source Names (DSNs) that point ODBC to a database.

Key Terms

Applications *(530)*
Boot Configuration Data (BCD) *(529)*
boot.ini *(526)*
bootmgr *(525)*
Component Services *(546)*
counter *(540)*
file association *(521)*
ntdetect.com *(528)*
ntldr *(525)*
object *(540)*
ODBC Data Source Administrator *(548)*
Performance *(536)*
Performance console *(539)*
Performance Logs and Alerts *(540)*
Performance Monitor *(542)*
Processes *(530)*
Registry *(519)*
Reliability and Performance Monitor *(542)*
root keys *(519)*
Services *(535)*
System Monitor *(540)*
Task Manager *(530)*
tasklist *(538)*

Key Term Quiz

Use the Key Terms list to complete the sentences that follow. Not all terms will be used.

1. The Windows XP boot process uses _____ to track all of the installed operating systems.
2. _____ is a command-line version of the Task Manager.
3. The _____ tab in the Task Manager displays overall CPU and memory usage.
4. The Performance console requires a(n) _____ and a(n) _____ to create logs about resource usage.
5. When you open the Registry Editor, you will see the five _____ on the left side of the window.
6. You can right-click on an item in the _____ tab of the Task Manager to see the associated services.
7. Windows 7 uses _____, instead of ntldr, to boot the OS.
8. The _____ tracks everything about your computer, from file types to user preferences.
9. You can open the _____ directly by pressing CTRL-SHIFT-ESC.
10. When programmers need to use objects across multiple applications, they turn to _____.

Multiple-Choice Quiz

1. Which of the following tabs is *not* found in the Task Manager?
   - **A.** Applications
   - **B.** Processes
   - **C.** Services
   - **D.** Objects
2. Which of the following root keys contains the data for a system's non-user-specific configurations?
   - **A.** HKEY_LOCAL_MACHINE
   - **B.** HKEY_USERS
   - **C.** HKEY_CURRENT_USER
   - **D.** HKEY_CLASSES_ROOT
3. What replaced the role of boot.ini in the Windows Vista/7 boot process?
   - **A.** Processes
   - **B.** ntldr

C. Boot Configuration Data

D. Data Sources

4. What is the name of the command-line command for killing tasks?

   A. kill

   B. taskkill

   C. processkill

   D. appkill

5. With what is every application and service in Windows associated?

   A. An object

   B. A counter

   C. A component

   D. A process

6. Which of the following are organized inside the Registry's root keys? (Select two.)

   A. Subkeys

   B. Subfolders

   C. Values

   D. Objects

7. Which of the following programs reads the boot.ini file to launch Windows XP?

   A. Reliability and Performance Monitor

   B. bootmgr

   C. ntldr

   D. Tasklist

8. Which of the following options are available when you right-click on a process in the Task Manager's Processes tab? (Select two.)

   A. Start Service

   B. Go to Service(s)

   C. Go To Process

   D. Set Priority

9. Mark wants to monitor a PC that's been having performance issues overnight. He needs something that will record the data while he's away from the computer. Which tool should he use?

   A. Performance Logs and Alerts

   B. Task Manager

   C. System Monitor

   D. Registry

10. Which of the following statements about booting in Windows 7 is true?

    A. BIOS does not use bootmgr.

    B. UEFI looks for the MBR, which finds the boot code that launches bootmgr.

    C. BIOS looks for the MBR, which finds the boot code that launches bootmgr.

    D. UEFI does not use bootmgr.

11. From which tab in the Task Manager can you set priority and affinity from a context menu?

    A. Services

    B. Applications

    C. Performance

    D. Processes

12. What is the name of the coding standard that enables programmers to write databases and applications that interact independently of the operating system used?

    A. Open Database Connectivity

    B. Services

    C. Component Services

    D. Reliability and Performance Monitor

13. How do you open the Registry Editor from the command prompt? (Select two.)

    A. regedit

    B. regedt32

    C. regeditor

    D. rgstry

14. Which of the following is *not* a system file necessary for booting Windows XP?

    A. ntldr

    B. boot.ini

    C. ntdetect.com

    D. bootmgr.efi

15. Which of the following key combinations opens the Task Manager?

    A. CTRL-ALT-SHIFT

    B. SHIFT-F5

    C. CTRL-SHIFT-ESC

    D. WINDOWS LOGO-PAUSE/BREAK

Essay Quiz

1. You work for a company that uses both Windows XP and Windows 7 machines. The Windows XP machines need to boot using BIOS, but the Windows 7 systems have been set up with UEFI. Your non-tech boss wants to know more about how Windows works. Explain to him the boot processes for both Windows XP and Windows 7 (with UEFI).
2. Your coworker Mike just discovered the Task Manager. He's been wreaking havoc on his applications, starting and stopping services, and killing processes left and right. You notice his behavior and decide to clue him in on how processes really work. Explain the differences and connections between applications, processes, and services.
3. Your boss wants you to help him disable several startup programs that have been slowing down his PC. He thinks the best way for you to do this is in the Registry. Before you begin your work, explain to him the importance and the dangers of the contents of the Registry.

Lab Projects

• Lab Project 15.1

The Task Manager is a powerful utility. Open it and take a look around. In the Applications tab, right-click on an application and select Go To Process. On the Processes tab, right-click on a process and select Go to Service(s). Explore these connections until you understand how it all fits together. This is also a great time to try ending a task or killing a process.

• Lab Project 15.2

Try tracking your CPU and RAM usage over time. Open the Performance tool in any current version of Windows and set up some objects and counters to monitor the information. How does this tool differ from the Task Manager's Performance tab?