

Part 1

Standard Windows Utilities

In This Section:

- ◆ Chapter 1: Using the Command Line Effectively
- ◆ Chapter 2: Completing Data-Specific Tasks
- ◆ Chapter 3: Discovering the System Status
- ◆ Chapter 4: Locating Files and Other Resources
- ◆ Chapter 5: Securing and Monitoring a System
- ◆ Chapter 6: Using Developer and Low-Level Utilities



Chapter 1

Using the Command Line Effectively

- ◆ Understanding Why the Command Line Is So Important
- ◆ Considering the Methods Available for Working at the Command Line
- ◆ Viewing the Commands by Purpose
- ◆ Updating Your Current Utilities at the Microsoft Download Center
- ◆ Configuring the Command Window
- ◆ Understanding Internal Commands
- ◆ Defining the Vista Command Line Differences

At one time, everyone worked at the command line. In fact, when you started the computer, you saw a command prompt and you never really left it the entire time you worked with the computer. I'm dating myself, of course, because no one's worked exclusively at the command line for many years. The days of DOS are gone and the command line is seemingly gone with it—or is it? The command line still exists and you can use it to make your life easier. In addition, working at the command line can help you automate tasks and work considerably faster. A good understanding of the command line can even help you work with fewer errors because most command line applications work or they aren't based on the input you provide. Of course, this begs the question of why people aren't using the command line if it's so great. This chapter answers that question; it helps you understand why the command line has fallen out of favor and why you should consider making it part of your life again.

Working at the command line doesn't mean that you have to perform tasks manually or memorize arcane syntax. It's true that you had to do that in the past to an extent, but even in the past, people created batch files so all they needed to remember was the batch file name and not the difficult series of command line switches for executing a command. Windows makes working at the command line a lot easier. You can even automate tasks so that you never actually go to the command line; you can tell Windows to perform all of that work for you. Consequently, working at the command line could mean putting a batch file together and then telling Windows to execute it for you. Working at the command line need not be time consuming or difficult.

Something to consider about the command line is that it contains a lot more than you might think. Many savvy administrators and power users know that Windows provides a number of command line utilities. However, few people realize just how many utilities there are. Would you believe that this book discusses 280 command line utilities of various types for all Windows users and a significant number more for Vista users? In fact, after performing the research for this book, I concluded that many of the most interesting Windows features aren't in the graphical user interface (GUI); they're at the command line. By the time you finish this book, you'll have gained an understanding of just how capable Windows is at the command line.

Understanding Why the Command Line Is So Important

You might have been there the day that Microsoft released Windows. The original reason for this product was twofold. First, it let users run more than one application at a time—something that required a kludge at the DOS prompt. Second, it provided a friendly interface that made using a computer easier. No longer did you have to remember command names; all of them appeared on screen so you could simply select the command you wanted to execute. The first version of Windows went over like a lead balloon, and the second version wasn't far behind, but by the third version, Microsoft had something workable—something people could use to perform their tasks without worrying about the command prompt.

Over the years Windows has delivered on its promise to make applications easier to use—at least the applications that you must sit in front of to use. For example, I wouldn't consider going back to a character mode word processor and I doubt very much that I'd want to write complex applications at the command line. Unfortunately, computing activities aren't limited to those tasks that you perform in real time in front of the display. Almost everyone has a task they must perform in the background or at least when they aren't present. The most common task that you should perform is backing up your data. Not only is there no need for you to be present when the backup occurs, but using your computer can be detrimental to getting a good backup because you should have all of the files closed. These noninteractive tasks always benefit from the command line because ease of use isn't an issue. When you perform a backup, you want it to be fast, accurate, and repeatable.

Okay, so you can count the number of tasks you need to automate on one hand? However, working at the command line can do a lot more for you than simply automate tasks that should take place in the background. Have you ever searched for text within a file using the Windows GUI and found that Windows Explorer can't locate text that you know appears within a certain folder? (Even with the advanced indexing features of Vista, you still can't find certain files because Vista doesn't index them and may not even provide direct access to them through the GUI.) Many people have found Windows Explorer lacking. Even when Windows Explorer can find the text, it isn't always accurate, and it's seldom fast. Interestingly enough, the command line offers utilities that can make searching for specific files quite fast and always accurate. For example, the FindStr utility discussed in Chapter 4 can help you locate text in any kind of file. You can even look inside binary files such as executables for particular strings. Everyone needs to search for data, and using the command line is usually faster than working with a GUI simply because the GUI gets in the way and slows things down.

Security has become a major issue with every cracker on the Internet seeking entry to your machine. However, have you ever wondered what's really running on your machine? You can't tell from the GUI. The best view you can get in most versions of Windows is Processes tab of the Task Manager that you can access by right-clicking the Taskbar and choosing Task Manager from the context menu. Vista adds a new Services tab that tells you about the services running on your system, but the addition only provides a little more help. Figure 1.1 shows the output from the Vista version of this application.

Unfortunately, Figure 1.1 shows only part of the story at best. For one thing, all of those SvcHost entries hide services that are running on your system (which is why that Services tab in Vista is so handy), which could be anything from the driver for your display adapter to a Windows service that is leaving you wide open to attack. However, you can't tell what's running on your system from Figure 1.1. Figure 1.2 shows the output of the TaskList command line utility. Suddenly you know about all of those SvcHost entries. As you can see, a single entry can host more than a few services. In addition, you now have access to a special number, the Process Identifier (PID). The PID lets you learn more about the application. In short, if you really want to know what your system is doing, you have to use the command line to do it. Don't worry too much about the TaskList utility right now; you'll find a discussion of its full capabilities in Chapter 5.

FIGURE 1.1
Task Manager only provides a partial view of the applications running on your system.

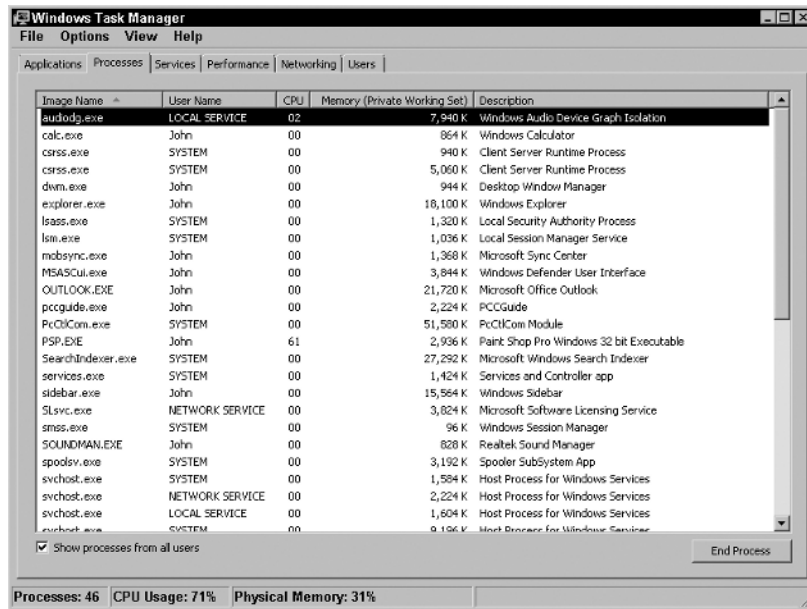
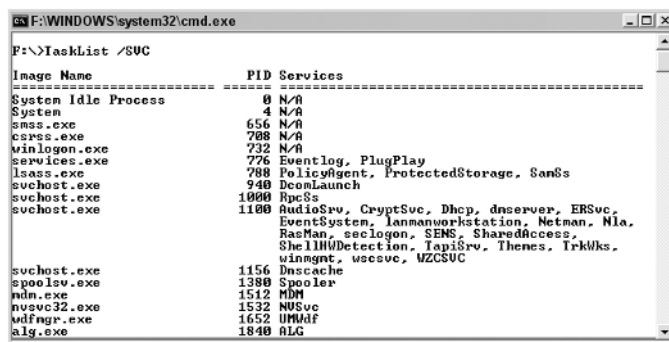


FIGURE 1.2
TaskList provides a complete picture of the applications running on your system.



The command line makes a wealth of powerful tools available. For example, you can discover the exact address for a Web site you visit frequently, so you can avoid making assumptions about emails that enter your inbox with an address, rather than human readable Web site name. On days when access to the Web sites you visit seems especially slow, you can use command line utilities to detect whether your local ISP is the problem or the problem is somewhere else that your Internet Service Provider (ISP) can't control before you call to complain. You can also use command line tools to locate local resources or those on a network. In fact, command line utilities can help you learn more about your system than you might think is possible.

The command line is important because it frees you from the constraints of the GUI that was supposed to make your life easier. Sure, you don't want to use the command line for everything, but it's good to know about the command line when you want to perform tasks quickly or you need low-level information about your system. The command line does require that you learn something about your machine, but this short section should have already demonstrated that you need the additional information the command line provides to keep your system safe and functioning fully.

The Command Line Made Easy

Some people are of the opinion that the command line works one way. You type in a command and hope that you got all of the information right and received the correct result, which you then have to interpret. This entire activity sounds quite difficult, somewhat boring, and error prone to say the least. You have to wonder why someone would put themselves through all that pain. However, the command line isn't anything like the scenario just mentioned. Actually, if you know a few simple rules, using the command line doesn't have to be hard at all. The following sections describe some of the methods you can use to work at the command line.

Using Utilities Directly

Generally, you'll be using the command line by working with the utilities directly. After all, it's a little hard to create a batch file or script if you don't know how the command works. However, using a command doesn't have to be hard. All you need to remember are two simple characters, `/?`. That command line switch says, "Help me!" The command usually will help by presenting you with some options for using it.

To open a command line, select the Start > Programs > Accessories > Command Prompt command. You'll see a command prompt. Whenever you open a command prompt using this method, it opens in your home directory on the hard drive. Type **TaskList** `/?` and press Enter. Figure 1.3 shows what you'll see. (I've scrolled back to the top so you can see the major entries.)

FIGURE 1.3

Make things simple; ask the command for usage instructions.

```

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
F:\Documents and Settings\John\TaskList /?

TASKLIST [/S system [/U username [/P [password]]]
          [/M [module] : [/SVC : [/U] [/FI filter] [/FO format] [/NH]]

Description:
This command line tool displays a list of application(s) and
associated task(s)/process(es) currently running on either a local or
remote system.

Parameter List:
/S system           Specifies the remote system to connect to.
/U [domain\user]   Specifies the user context under which
                   the command should execute.
/P [password]       Specifies the password for the given
                   user context. Prompts for input if omitted.
/M [module]         Lists all tasks that have DLL modules loaded
                   in then that match the given pattern name.
                   If the module name is not specified,

```

The first piece of information is the usage instructions for the command. A set of square brackets (`[]`) tells you about an optional input. In this case, everything is optional; you can use `TaskList` by itself.

A slash (`/`) tells you about a command line switch. Sometimes command line switches appear with a dash (`-`) instead. In either case, a command line switch configures the command to perform a task in a specific way. For example, `TaskList` doesn't normally display services, but you can tell it to display services by adding the `/SVC` command line switch.

Some command line switches depend on other command line switches. You'll see the command line switches nested within multiple layers of square brackets in this situation. For example, if you want to supply a password for logging into a remote system to view the tasks running on it, you must also supply the `/System` and `/Username` command line switches.

In other cases, command line switches are mutually exclusive. The command line will separate these switches with the pipe (`|`) symbol. The `TaskList` command won't allow you to use the `/M` command line switch with the `/SVC` switch; you must select one or the other.

After the usage information, you'll normally see a description section for newer commands. The description tells you what task the command performs and why you would want to use it. Sometimes this information is quite complete, as it is with the `TaskList` command, and in other cases, you'll still

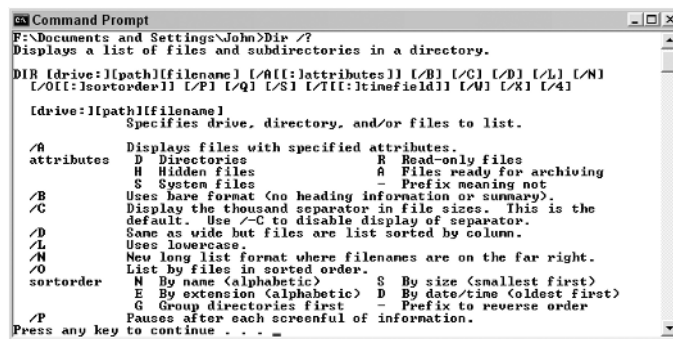
be scratching your head after you read the description. Older commands don't provide a description at all; you just have to know what task they perform, which is why many people don't use them.

A description of the individual parameters (or arguments and inputs) comes next. These entries tell you how to use the individual command line switches. You'll also discover other kinds of information you must provide. For example, the `Dir` (directory) command information shown in Figure 1.4 tells you that you can provide a drive letter, followed by a colon, followed by a directory path, and ending with a filename specification. None of these entries is a command line switch, but they're all important parameters.

The final section is a list of examples. Only a few commands provide this kind of information, but it's always helpful when they do. The examples come in many forms. The `TaskList` command provides a list of filters first, so you can see how to get the output you want. It provides actual usage examples next so you can see what to type at the command line. The point is that most people could use a command at the command prompt if they simply knew the simple `/?` command line switch. Go ahead and try it out now with the `TaskList` and `Dir` commands. You'll want to keep the `/?` command line switch in mind as you read about other commands in this book. Try it out with every one of them and you'll find that most commands provide some information, usually enough to jog your memory when you need to use it.

FIGURE 1.4

Sometimes you provide text input as well as command line switches.



```

F:\Documents and Settings\John>Dir /?
Displays a list of files and subdirectories in a directory.

DIR [drive:][path][filename] [/A[:attributes]] [/B] [/C] [/D] [/L] [/N]
[/O[:sortorder]] [/P] [/Q] [/S] [/T[:timefield]] [/W] [/X] [/4]

[drive:][path][filename]
    Specifies drive, directory, and/or files to list.

/A
    Displays files with specified attributes.
    attributes  D Directories          R Read-only files
               H Hidden files         A Files ready for archiving
               S System files         - Prefix meaning not

/B
    Uses bare format (no heading information or summary).

/C
    Display the thousand separator in file sizes. This is the
    default. Use /-C to disable display of separator.
    Same as wide but files are list sorted by column.

/D
    Uses lowercase.

/N
    New long list format where filenames are on the far right.

/O
    List by files in sorted order.
    sortorder  N By name (alphabetic)  S By size (smallest first)
               E By extension (alphabetic)  D By date/time (oldest first)
               G Group directories first - Prefix to reverse order

/P
    Pauses after each screenful of information.

Press any key to continue . . . =
  
```



Real World Scenario

STORING COMMANDS IN BATCH FILES

I've worked at the command line for years, so you might assume that I have all of these commands memorized by now. However, like many people, I find that memorizing all of the commands, their parameters, and their command line switches is just too much work. However, discovering the required parameters one time isn't too much work. That's where batch files come to my aid. I use batch files to remember specific command sequences for me.

When you need to store one or more commands so you don't have to remember them every time you want to use them, a batch file can do the job. In fact, you can create batch files that have a limited amount of intelligence so they don't perform the same task in the same way every time. Batch files are the first method that many people use to automate the command line. I have batch files that I wrote over 18 years ago when I started with computers and I'm still using them today. In short, a good batch file can last a very long time. The thing to remember about batch files is that they're very easy to write, only have a little intelligence (so there isn't any heavy coding), and don't require anything special to execute. You'll discover how to work with batch files in Chapter 7.

Writing Scripts

Scripts are the next step up in complexity. A script uses a simple programming language to accomplish tasks. You can't create complex applications using a script. For example, you wouldn't want to write a word processor using a script. However, scripting languages provide more intelligence than a batch file can. In addition, you can access some of the functionality that Windows provides. Consequently, rather than rely on utilities for every action, you can ask Windows for some help in automating your tasks.

A script requires a special environment to run. Windows provides this environment in the form of a script interpreter. The interpreter reads every line of code you write in your script and performs the task it requests. Writing scripts is a little harder than writing batch files, but not nearly as difficult as writing an application with a full-fledged programming language. Consequently, scripts are exactly what many people need to automate tasks when they don't want to learn a full-fledged programming language, yet find batch files less robust than they'd like. You'll discover how to work with scripts in Chapter 8.

Most of the tasks you perform using scripts have standard requirements and needs to execute successfully. Active Directory, the Windows enterprise database, requires some special handling to work correctly. Chapter 9 discusses the scripting requirements for this special environment and helps you create scripts that make working with Active Directory a lot easier.

Scheduling Tasks

No matter how you work with the command line, whether you use individual commands, batch files, or scripts, you can schedule a task to run at a specific time. For example, if you want to defragment your hard drive every night, you can schedule the Defrag command described in Chapter 6 to run automatically. Of course, you'd better be certain that everything is set up correctly before you assume the computer can perform the task on its own. Many people begin using the Task Scheduler to run tasks that they could forget during normal work hours and then progress to after-hours tasks. You'll find a discussion of the Task Scheduler in Chapter 10 and after-hours task scheduling in Chapter 11.

Relying on Third Party Utilities

The fact that Microsoft doesn't spend much time advertising the command line should tell you something. The tools that Microsoft provides for working at the command line are basic, simple, and not always the best tools at your disposal. Third party tools for working at the command prompt have been around for a long time. Most of these products are mature, fully tested, and quite capable of making your command line experience everything it should be. Part 3 of this book, Chapters 12, 13, and 14, provides you with a wealth of third party utility resources.

Viewing the Commands by Purpose

The commands on your system have a particular purpose in most cases. The name doesn't always reveal the purpose. Depending on the documentation provided with the utility, you might still have a hard time figuring it out. However, they all do have a particular purpose. For example, the `Dir` command helps you locate files and directories (folders) on your machine and the `TaskList` command helps you discover which applications are running. The `Dir` command performs a data-specific task, while the `TaskList` command is a monitoring application. The following sections describe the classifications of commands that you'll find at the command prompt.

Data Specific

Many of the commands that Windows provides are data specific. You use them to perform infrastructure tasks such as creating and removing directories. Other commands help you create, delete, and edit files. You'll find that the `Sort` command lets you sort the contents of a file. Some of the commands display data on screen, while others send the file content to the printer. A few of the commands perform management tasks. For example, you can perform a bulk copy of your files using the `XCopy` command. All of these commands appear in Chapter 2.

The data-specific commands are important for a number of reasons. For example, you can write a batch file that lets you set up the entire directory structure for a new user. A new user setup can require seconds instead of hours. In addition, you can be certain that every user will have precisely the same setup every time, which means that you'll spend less time supporting a network and more time getting other work done.

Using the data-specific commands can save you considerable time in other ways. Most companies archive files either when a project finishes or during standard intervals in the process of working with a client. Batch files can make it significantly easier to create the archive, but you need to know the commands required to create the data infrastructure and move the files first.

System Status

Computer systems today are very complex. The combination of software and hardware that makes the computing environment as useful as it is can also hide problems and eventually damage the very data they were used to create. Knowing the status of your system is important. However, discovering the status information can be hard without the use of the command line. Something as simple as knowing what equipment you have installed can make a big difference when it comes time to manage the system. Chapter 3 tells you all about the system status commands.



Real World Scenario

USING STATUS INFORMATION TO YOUR ADVANTAGE

Failures of any kind on a computer can prove frustrating. The question of where to start looking for the problem can be the first and last question that many people ask. It's too easy to see the computer as a box that has a problem and assume there isn't any place to look. I've talked more than a few people through computer problems by simply telling them about the status indicators that the computer provides. In many cases, Microsoft provides these commands as a means for their support staff to locate a problem for you at some outrageous hourly rate, but there isn't any reason you can't use the tools too.

For example, one command problem that people encounter is a failed audio system. You can check the event log and then view the information about the sound system using the Control Panel applets. In addition, you can use a utility such as `DXDiag` to perform audio checks on your system. You might even use performance monitoring to look for hidden audio problems. Of course, you have to remember to do all of these things. However, as the book progresses, you'll find that you can also access all of this information from the command prompt. A batch file might be all you need to perform a carefully executed diagnostic check using the same steps every time. The results are consistent input about your audio system and no missed checks, which means that you have a good chance of locating an error without paying anyone.

Of course, you don't want to spend all of your time managing the system and obtaining the status information. You can also use the command line to set up performance monitors, alerts, and logs. Of course, you can use the Performance console in the Administrative Tools folder of the Control Panel to perform the required setups, but that means performing the task manually. If you have more than one computer to manage, it's a lot easier to set up a script or batch file to perform the required setups once and then automate the task on every machine you manage.

TIP If you think that utilities such as DXDiag (DirectX Diagnostics) require use of a GUI to perform any useful work, be prepared for a surprise in Chapter 3. Many of these utilities sport a command line interface that you can use to manage computers from your desk, rather than running from place to place looking for information. A GUI is great when you're sitting at the machine because it does make things easier, but the command line interface makes things faster and more convenient when working from a remote location.

File and Resource Management

Files and other resources are always a source of concern for a computer system. The resources you have at your disposal determine the kind and amount of work you can perform with the computer system. Data isn't simply a collection of information that you use to create a report, it's a resource that you have to manage. The utilities in Chapter 4 all provide some type of resource management. The chapter begins by looking at file commands, such as those you can use to detect strings within a file, but it also includes other resources. For example, this chapter shows how to manage the power configuration settings on a system from the command prompt. You'll also find commands for a number of services including the Remote Access Server (RAS).

Security and System Monitoring

Monitoring on a computer can take several forms. There's the kind of monitoring that you perform to ensure the computer is operating at peak efficiency that appears in Chapter 3. However, if you only check the performance of your computer, you'll almost certainly notice that it decreases with time. The reason is simple: overall computer health is a combination of performance, reliability, and security. Often, you increase one part of the triangle at the expense of the other two. For example, increasing the performance of the computer by overclocking the hardware will almost certainly result in reduced reliability and could impair security as well.

Chapter 5 focuses on the kind of monitoring that improves security from a number of perspectives. The security monitoring in this chapter doesn't necessarily keep intruders out, but it focuses on the kind of monitoring that dissuades outside intrusion and improves system health. For example, by maintaining strict control over the network, you not only improve overall system security but you also improve the performance and reliability of the computer as well.

Developer and Low-level Tasks

Microsoft has always tried to provide support for the developer community by including helpful utilities for them as part of Windows. For example, after you install a program, the developer can use the ShutDown utility to restart your system and ensure that the changes to system DLLs load. However, developers aren't the only ones to use this utility. I include a quick shutdown feature for my system using a simple shortcut as explained in the "Shutting the System Down with the Shut-Down Utility" section of Chapter 6. Using this simple shortcut shortens a relatively long shutdown process into one that takes seconds (sometimes less). Of course, you have to know when to use and when to avoid a quick shutdown.

Most of these low-level utilities work with the system in ways that could be dangerous in the wrong hands. Consequently, you'll want to view this chapter only if you have the skills required to work with system resources safely. For example, the DiskPart utility could wipe out your hard drive, so it's important that you not use it unless you understand disk partitioning.

Active Directory

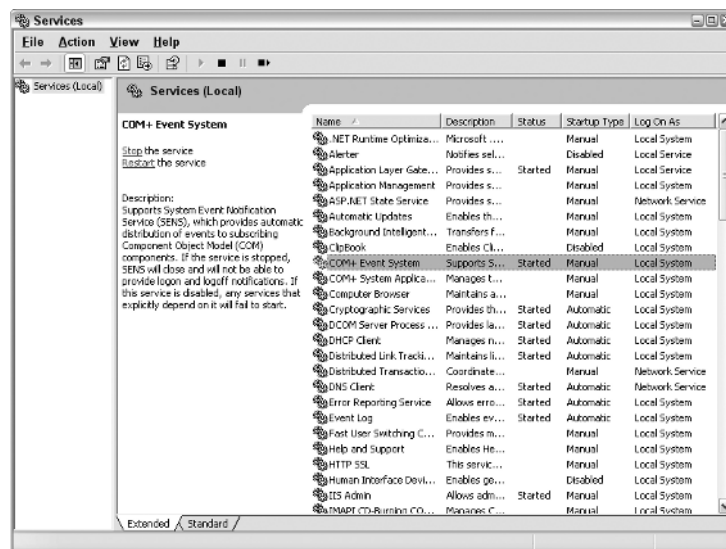
Some tools are specifically for managing enterprise systems. All of the Active Directory utilities described in Chapter 9 fall into this category. In fact, because of the nature of these utilities, the chapter describes them in the light of usage with scripts immediately, rather than assuming you'll use the utilities in a stand-alone mode. If your company uses Active Directory, this chapter can save you significant time and frustration, while making your setup considerably more reliable.

Services

Windows services are a special breed of applications. In fact, many people ignore them completely. However, services are simply a kind of application, one that executes in the background unobserved, often waiting for a special system event to occur. If you haven't really paid attention to services before, you can view them using the Services console located in the Administrative Tools folder of the Control Panel. Figure 1.5 shows a typical view of services.

Unfortunately, failure to manage services can cause all kinds of problems. For example, every service uses system resources, so keeping a service that you don't need running can slow system performance. Some services, such as Messenger (not associated with Windows Messenger), can actually open security holes in your system. By using command line utilities combined with batch files, you can start and stop services as you need them. For example, I start the development-oriented services on my system only when I plan to develop code; the rest of the time, I keep them disabled so they don't use resources or open security holes. Starting and stopping is a matter of double-clicking a simple batch file, which makes it incredibly easy to maintain a secure and efficient environment.

FIGURE 1.5
Managing services is an important reason to use command line utilities.



WARNING The names and order of services can vary by Windows version. For example, Figure 1.5 shows the Windows XP names of the services. In most cases, these names are unchanged from Windows 2000 and remain the same in Windows 2003. Vista makes the largest changes to service names and even old favorites have new names. Because you need the actual names of services to use some command line utilities, you'll want to verify that any older batch files that manipulate services still work when you move them to a new version of Windows.

Task Scheduling

It's not always convenient to run commands while you're using the computer for work. In other cases, you want to ensure the command runs even if you get busy in meetings. You can resolve both needs by using the Task Scheduler. Chapter 10 tells you how to use the Task Scheduler to improve the efficiency of your system, while Chapter 11 provides a special focus on after-hours scripting using the Task Scheduler. In both chapters, you'll discover new techniques for using command line utilities to control the Task Scheduler so it performs as you expect.

Updating Your Current Utilities at the Microsoft Download Center

You might already know about the Microsoft Download Center at <http://www.microsoft.com/downloads/search.aspx>. If you don't, you should visit it before you go any further in the book. This Web site provides access to updates for all of Microsoft's products, including those that the Windows Update and Office Web sites don't automatically update for you.

The Microsoft Download Center usually displays the current favorite or target applications at the top. If you visit the Web site often, you'll want to check out this list immediately to obtain the current versions of applications you already have installed on your system.

Immediately below the list of favorites (you usually have to scroll down), you'll see a search form where you can search for applications by keyword and technology. In many cases, the most efficient search is to look for applications by technology because Microsoft sometimes uses arcane terminology for the updates.

TIP If you really have a hard time finding an application you need at the Microsoft Download Center, go to Google Advanced Search at http://www.google.com/advanced_search. Type the name of the product you want to find in the With All of the Words field. Type the www.microsoft.com domain in the Domain field. Click Google Search and you should find the application you need with relative ease.

The final section of the Microsoft Download Center contains download categories. Use these links when you have an idea of what you need, but don't know the name. As an example, you might have heard about something interesting on a newsgroup, but might not know precisely what Microsoft calls it.

Configuring the Command Window

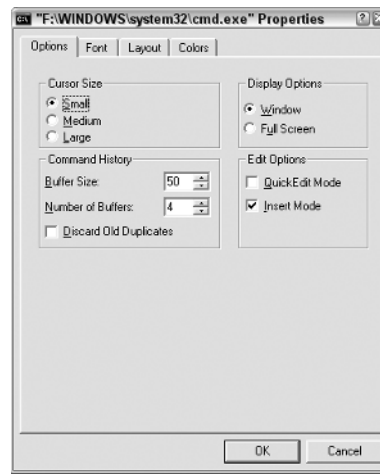
Many users start the command window, see the typical command prompt, and just assume that they'll never see anything else. However, you can easily configure the command window to appear as you want, at least within limits. You can access these features by clicking the box in the upper left corner and choosing Properties from the context menu. You'll see a properties dialog box with four tabs. Each of these tabs is described in the sections that follow.

Setting the Window Options

The Options tab shown in Figure 1.6 defines how the command window reacts when you open it. The Cursor Size option controls the size of the cursor, with small being the default. The Large option provides a block cursor that is very easy to see. The Display Options determine whether you see the command window full screen or as a window. Using the full screen mode when you have a number of tasks to perform is easier on the eyes.

FIGURE 1.6

The Options tab helps you control the appearance and behavior of the command window.



NOTE Vista doesn't let you run the command window in full screen mode by changing the Display Options setting. This particular option is missing when you view the dialog box shown in Figure 1.6. However, you can set the option by changing the command line prompt shortcut options. If you want to use full screen mode all of the time, right-click the Command Prompt entry in the Start ► Programs ► Accessories menu and choose Properties. Select the Options tab of the Command Prompt Properties dialog box and select Full Screen in the Display Options group.

The Command History is especially important. The Buffer Size option determines the number of commands the buffer will store. Every command requires memory, so increasing this number increases the amount of memory the command prompt requires. Increase this number when you plan to perform a number of complex commands. A smaller number will save memory for larger command line applications. The Number of Buffers Option controls the number of individual histories. You need one history for each process (application environment) you create. Generally, the four shown work fine.

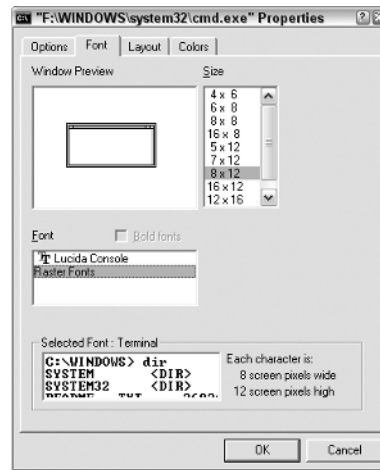
The Edit Options determine how you interact with the command window. Check the QuickEdit Mode when you want to use the mouse to work with the entries directly. The only problem with using this feature is that it can interfere with some commands such as Edit that have a mouse interface of their own. The Insert Mode option lets you paste text into the command window without replacing the text currently there. For example, you might copy some information from a Windows application and paste it as an argument for a command.

Changing the Font

The Font tab shown in Figure 1.7 controls the font used to display text. The font size automatically changes when you resize the window, but you can also control the font size directly using this tab. The raster fonts give the typical command line font appearance that works well for most quick tasks. The Lucida Console font works better in a windowed environment. It's easier on the eyes because it's smoother, but you might find that some applications won't work well with it if they create "text graphics" using some of the extended ASCII characters. The extended ASCII characters include corners and lines that a developer can use to draw boxes and add visual detail.

FIGURE 1.7

Use the Font tab to control the size of the text in the command window.



Choosing a Window Layout

The Layout tab shown in Figure 1.8 has the potential to affect your use of the command window greatly when working in windowed mode. The Screen Buffer Size controls the width and height of the screen buffer, the total area used to display information. When the Window Size setting is smaller than the Screen Buffer Size, Windows provides scroll bars so you can move the window around within the buffer area and view all it contains. Some commands require a great deal of space for display purposes. Adjusting the Screen Buffer Size and Window Size can help you view all of the information these commands provide.

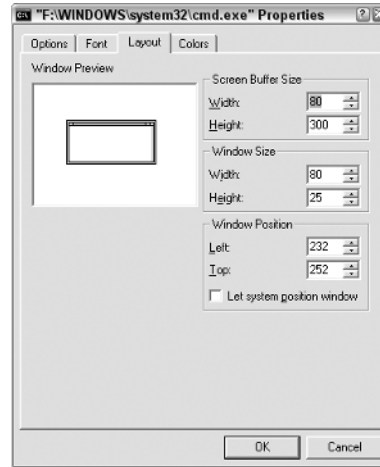
The Window Position determines where Windows places the command window when you first open it. Some people prefer a specific position on the screen so they always know where a new command window will appear. However, it's generally safe to check Let System Position Window to allow Windows to place the command window on screen. Each command window will appear at a different, randomly chosen, position on screen.

Defining the Text Colors

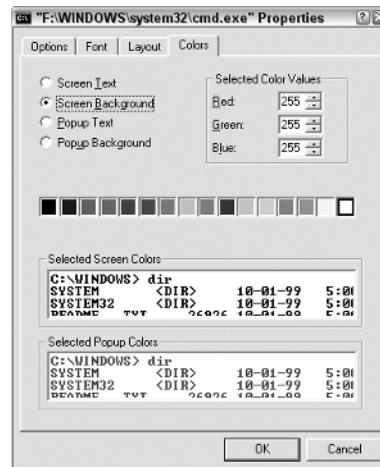
Microsoft assumes that you want a black background with light gray letters for the command window. Although DOS used this setting all those years ago, many people today want a choice. The Color tab lets you choose different foreground, background, and pop-up colors for the command window (even though Figure 1.9 doesn't show the colors, it does present the dialog box layout). You can modify the window to use any of the 16 standard color combinations for any of the text options. Use the Select Color Values options to create custom colors.

FIGURE 1.8

Change the size and positioning of the command window using the Layout tab.

**FIGURE 1.9**

Modify the text colors for an optimal display using the Colors tab.



Placing a Command Prompt at Your Fingertips

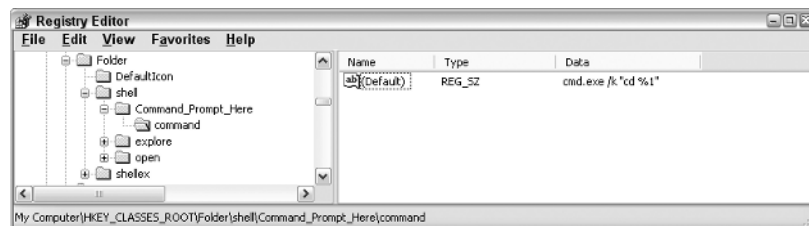
It's possible to change the Windows Explorer registry settings so that you can get a command prompt wherever you need one. For example, say you're viewing the System32 folder and see a utility that you've never seen there before. You can use this registry change to right-click the folder and choose Command Prompt Here from the context menu to see a command prompt in that folder, rather than your home folder, as normal. Use the following steps to make this change manually.

1. Select the Start ➤ Run command. Type **RegEdit** in the Open field and click OK. You'll see the Registry Editor.

NOTE When working with Vista, you'll often see a User Account Control (UAC) dialog box appear that asks whether you really intend to perform a particular action. Although this dialog box can become quite annoying, it does serve a useful purpose in alerting you to actions from viruses and other nefarious applications. Whenever you see the UAC dialog box and know that you've started a particular action, simply click Continue and Vista will continue the action (assuming you have the proper rights). See the "Vista Changes for the Command Line" section of the chapter for more details.

2. Open the HKEY_CLASSES_ROOT\Folder\shell folder. Right-click this folder and choose New ➤ Key from the context menu. The Registry Editor will create a new type for you.
3. Type **Command_Prompt_Here** as the key name and press Enter.
4. Right-click the Command_Prompt_Here key and choose New ➤ Key from the context menu. Type **command** for the new key name and press Enter. You now have two new keys, as shown in Figure 1.10.
5. Right-click the command key and choose New ➤ String Value from the context menu. Type **cmd.exe /k \"%cd %1\"** as the new string value. Exercise extreme care with this step. Press Enter. The new value should look like the one shown in Figure 1.10.
6. Close the Registry Editor.

FIGURE 1.10
Create new registry keys to hold the Command Prompt Here context menu option.



Congratulations, you now have a tool that you can use to create a command prompt directly from Windows Explorer. Open a new copy of Windows Explorer, right-click a folder, and you'll see the new Command Prompt Here entry. Select this option to create a new command prompt in the folder that you right-clicked. This is the first use of the command line in this book. You can learn more about CMD.EXE in the "Using the CMD Switches" section of Chapter 7.

You don't have to go through this set of steps every time you want to add this feature to a copy of Windows. The following registry script will perform the same task. To use this approach, open a copy of Notepad and type the script shown here precisely, as shown.

Windows Registry Editor Version 5.00

```
[HKEY_CLASSES_ROOT\Folder\shell\Command_Prompt_Here]
@="Command Prompt Here"
```

```
[HKEY_CLASSES_ROOT\Folder\shell\Command_Prompt_Here\command]
@="cmd.exe /k \"%cd %1\""
```

When you finish, select the File ➤ Save command. Type **CommandPromptHere.REG** in the File Name field. Choose All Files in the Save as Type field. Click Save. You now have a new registry script for adding the Command Prompt Here feature. All you need to do is double-click this file in Windows Explorer to make the addition.

Understanding Internal Commands

This chapter has used the term *command* for everything you execute at the command line. In reality, you need to view the command line as having multiple command types. Some commands, such as `TaskList.EXE`, appear as separate files. This book will use the term *utility* for these kinds of commands from now on. A utility always resides in a separate file and you can look it up using the `Dir` command.

Some commands don't exist in separate files; they reside in the host program that you use to interact with the computer. The host program for the command prompt is `CMD.EXE`. If you want to try it out, use the Start ➤ Run command to display the Run dialog box. Type **CMD** in the Open field and click OK. You'll see a command prompt. `CMD.EXE` doesn't end after it opens the command prompt; it remains in the background to receive and react to your keystrokes.

The `CMD.EXE` file also has a number of internal commands. These special keystrokes tell `CMD.EXE` to perform a task for you. For example, the `Dir` command is an internal `CMD.EXE` command. You won't find `Dir` listed as an executable anywhere on your hard drive. This book lists all internal commands as commands. Consequently, you'll see the `TaskList` utility and the `Dir` command discussed later in the book.

Other utilities create a host environment and you'll discover the commands in those host environments as you read the book. For example, the `TelNet` utility discussed in Chapter 4 provides a host environment where you'll type commands. These commands don't exist outside `TelNet`, just as the `Dir` command doesn't exist outside of `CMD.EXE`.

Vista Changes for the Command Line

Microsoft is well known for maintaining backward compatibility whenever possible despite a strong desire to add new features to an operating system or application that serve to complicate administrative tasks. However, you're going to find that Vista represents a change in tactic. Everyone has complained for so long about the security problems in Windows that Microsoft has finally decided to do something about the issue (proving yet again that you should be careful about what you wish for). Working at the command line is considerably harder in Vista than in any previous version of Windows, partly because of the new security features and partly because of significant changes to some command line features. The following sections provide an overview of these various changes and help you understand them better.

Understanding User Account Control (UAC) Changes

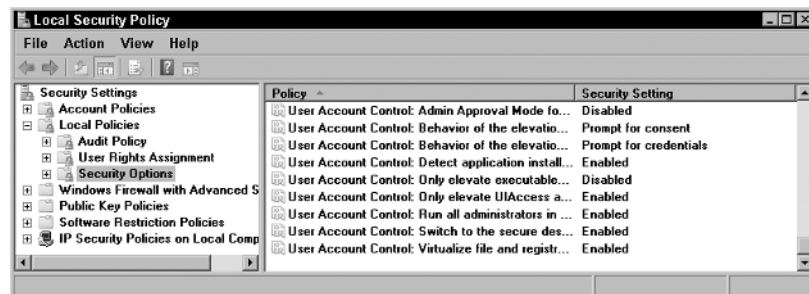
The UAC features of Windows serve to increase security by reducing the chance that an application can perform any act on the user's behalf, without the user's knowledge. Vista assumes that every user is a standard user, even administrators. If you have an administrator account, you must elevate your privileges from standard user to administrator to perform many tasks. In most cases, this means clicking Continue when you see the UAC dialog box asking whether you really mean to perform a particular task. Let's just say that the feature is incredibly annoying for anyone who spends their day working at the command line, but it does serve a useful purpose. Used correctly, UAC ensures that no one can perform an action on your behalf without your knowledge. Given that administrators have considerable power, this feature is especially useful to administrators who might become targets of nefarious individuals. After all, you don't want to suffer the embarrassment of being the source of a virus, adware, or spyware on the very network that you're supposed to protect.

However, UAC goes far further than simply asking whether you want to perform a particular task. In some cases, it can actually prevent you from performing tasks despite having an administrator account. For example, you're going to find that Vista severely hampers your access to the Windows and System32 folders even with an administrator account. Vista meets any attempt to change anything in the folders with disapproval that is seemingly impossible to overcome. The same holds true for the root directory of the boot drive. Network drives are nearly impossible to access as well. In fact, except for your personal data folders, Vista is locked down so tight that many administrative tasks are all but impossible to perform, even with an administrator account. The "Overcoming UAC Problems," "Giving Yourself Permission," and "Setting Vista Zones on Network Drives" sections of the chapter provide you with details on how you can overcome some of these issues. The bottom line is that Vista is all about security. Microsoft has thrown backward compatibility out the window in order to achieve some level of additional security.

Overcoming UAC Problems

The main source of woe for most administrators in Vista is the UAC. Before you can do anything, you'll need to override the UAC, at least for a while. It's actually better if you can override UAC to gain the privileges you need and then return Vista to its default state. Fortunately, you'll find all of the UAC controls in one place. Open the Local Security Policy applet found in the Administrative Tools folder of the Control Panel. Select the Security Settings\Local Policies\Security Options folder as shown in Figure 1.11. The figure shows the default settings should you ever need to restore them.

FIGURE 1.11
The User Account Control policies affect your ability to work with the system as an administrator.



The best way to get your system configured is to turn off everything, reboot the system, and make the changes you need (see the "Giving Yourself Permission" and "Setting Vista Zones on Network Drives" sections for details). After you set the required permissions, return Vista to its previous state so you can obtain the security features that Vista provides. Remember that changing the settings isn't enough since a change to policy won't affect your security token until after you reboot. Always make the required changes and then reboot to add those changes to your security token.

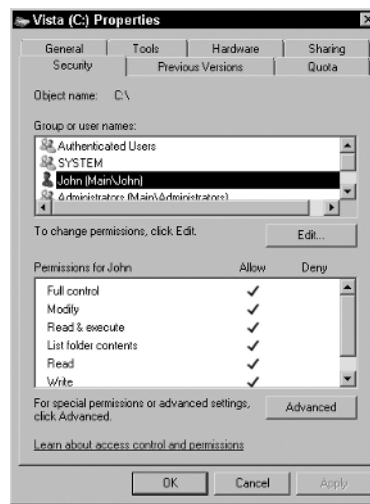
Giving Yourself Permission

Vista handles security differently from previous versions of Windows. First, you don't have permission to use anything other than your personal folder until you specifically request that permission. If UAC is still in force, Vista will reject any request to change security in the root folder of the boot drive (the one that has the Windows folder) even if you have an administrator account. Consequently, you must disable UAC to give yourself permission to access the drive.

Second, you'll quickly discover what it means to be a standard user. Whenever you open Windows Explorer, it opens with standard user credentials, which means you can't change anything even if you have administrator privileges and disable UAC. Instead of starting Windows Explorer normally, you'll need to right-click Start > Programs > Accessories > Windows Explorer and choose Run as Administrator from the context menu. This action gives you a version of Windows Explorer with administrator privileges. Now, you can right-click the root directory and choose Properties from the context menu. Select the Security tab and add your account to the list as shown in Figure 1.12. Adding a group won't grant you the privileges you think it will—you must add your personal user account to the list.

FIGURE 1.12

Grant yourself permission to make changes to your own hard drive.



Third, you'll be amazed to find that rights don't flow as they once did. Microsoft has specifically blocked the flow of rights from the root directory to the Windows and System32 folders. Consequently, you must also add your account to these folders if you want to access them as shown in Figure 1.12. Rights do flow from the root folder to other folders on the boot drive.

Setting Vista Zones on Network Drives

Vista treats every network drive as an Internet drive. Choose View > Status Bar in Windows Explorer and you'll notice that the status bar shows that all network drives are now in the Internet zone, which severely limits what you can do with them. The purpose of this change is to make it harder for viruses, adware, and spyware to spread from machine to machine. However, it also makes it nigh on to impossible for an administrator to perform any task remotely. Unless you want to spend your days running from machine to machine, you need to change the zone of those network drives.

Admittedly, you'll only want to change the zone for drives that you actually work with regularly. You may even want to elevate their privileges when working with a special account. It depends on how much you value this particular Vista feature. The following steps help you set the zone for your network drives.

1. Choose a network drive. Double-click the zone icon in the Windows Explorer status bar. You'll see the Internet Security Properties dialog box.

2. Choose the Trusted Sites zone and click Sites. You'll see the Trusted Sites dialog box.
3. Type the URL for the network machine, such as `file://winserver` when the machine's name is winserver. Click Add. The new location appears in the trusted zone list.
4. Click Close to close the Trusted Sites dialog box and OK to close the Internet Security Properties dialog box. Your network drive is now accessible as it was in previous versions of Windows.

Understanding Vista Doesn't Support Old Commands

Vista changes some of the command line commands that have existed since the days of DOS. For example, you're going to notice a change in the `Choice` command (described in the "Using the Choice Command" section of Chapter 7). Many of the changes are obviously for security reasons, but changes to commands such as `Choice` simply end up breaking batch files for no apparent reason. Microsoft hasn't offered any reason for many of these changes, but you'll find them all listed as the book progresses.

Some commands are missing entirely. Microsoft chose not to support some commands that are obviously long in the tooth and it isn't hard to understand why. Some commands have simply become dangerous to use because they work completely different from other commands and because very few people use them. However, some command changes occur because of the way Vista operates. For example, Microsoft has replaced the `BootCfg` utility with the `BCDEdit` utility because Vista uses an entirely different method to store the boot configuration. You'll also see these changes listed throughout the book.

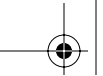
The bottom line for anyone using Vista is that you need to test every batch file to determine whether Vista changes have modified their behavior. Even simple batch files are prone to break in the new environment. Look for the changes in the chapters of this book whenever you have a question about some new Vista behavior.

TIP Don't always assume that a Vista command or utility change has affected your batch file. In some cases, batch files will cease to work because you don't have the required permissions. Always try to start the batch file by right-clicking it and choosing **Run as Administrator** before you assume that the command or utility won't work. Remember that security is everything in Vista.

Getting Started with Command Line Tasks

If you're anything like me, you're a little overwhelmed by now at what the command line can do for you. I've always used the command line. In fact, I've had some batch files hanging around since the days of DOS—yes, really, that long. However, until you take time to look at what the command line has to offer, you don't know what's there. Microsoft certainly doesn't make the command line the centerpiece of its advertising. In fact, the command line is one of the least understood and explored parts of Windows. Consequently, this book is your doorway to a new world. Not only will you perform tasks faster, with less effort, and more precisely, but you'll have a distinct edge over those around you as well. While they fiddle with an excessively time-consuming GUI, you're speeding along at the command line and making yourself look quite good in the process.

Of course, before you can begin working at the command line, it pays to make sure that your system is ready. Before you go any further, make sure you get on Windows Update and download all of the latest patches for your system. Check your Office installation and all of those third party utilities as well. Go to the Microsoft Download Center and look around for the downloads you've missed. An updated system normally yields the best set of fully updated utilities that will perform best and with the fewest possible errors. Make sure you get your command line prompt set up and add the required



registry entries to put a command prompt at your fingertips as well. If you're using Vista, make sure you follow the procedures in the "Vista Changes for the Command Line" section of the chapter to set up your system for command line use in Vista. If you don't perform this setup, you'll find that most of the commands and utilities in this book won't work at all.

Chapter 2 begins showing you the command line utilities. It focuses on commands that affect data in some way. You'll discover how to work with files and directories. It also shows some of the command line editors at your disposal. You don't have to use these editors, but they can help you create batch files and perform other tasks that make your command line experience better. Chapter 2 also provides your first look at the registry and some of the productivity utilities that you'll script later in the book. For example, it describes how to use the backup utility at the command line.

