CHAPTER 14

Using the Command Prompt

The command prompt provides technicians with another way to accomplish tasks. Many end users can get by without ever using the command prompt. However, it provides a wealth of tools for technicians and administrators and is extremely valuable. When you're taking the live exams, don't be surprised if you come across a performance-based question that requires you to use the command prompt. Instead of an answer consisting of multiple choices, you might see a command prompt. You'll need to enter the appropriate commands to answer the question correctly.

220-802 Exam objectives in this chapter:

- 1.3 Given a scenario, use appropriate command line tools.
  - Networking
    - PING
    - TRACERT
    - NETSTAT
    - IPCONFIG
    - NET
    - NSLOOKUP
    - NBTSTAT
  - OS
    - KILL
    - BOOTREC
    - SHUTDOWN
    - TLIST
    - MD
    - RD
    - CD
    - DEL

        - FDISK
        - FORMAT
        - DISKPART
        - CHKDSK
        - COPY
        - XCOPY
        - ROBOCOPY
        - SFC
        - [command name] /?
- 1.4 Given a scenario, use appropriate operating system features and tools.
    - Run line utilities
        - MSCONFIG
        - REGEDIT
        - CMD
        - SERVICES.MSC
        - MMC
        - MSTSC
        - NOTEPAD
        - EXPLORER
        - MSINFO32
        - DXDIAG
- 4.3 Given a scenario, troubleshoot hard drives and RAID arrays with appropriate tools.
    - Tools
        - CHKDSK
        - FORMAT
        - FDISK
- 4.6 Given a scenario, troubleshoot operating system problems with appropriate tools.
    - Tools
        - Sfc

REAL WORLD **BECOME A GREAT ADMINISTRATOR**

A sentence many students hear me say in the classroom is, "The difference between a good administrator and a great administrator is the ability to script." Scripts allow you to automate tasks and complete complex tasks quickly, and they make you look like an IT rock star.

What's this have to do with the command prompt? Scripting starts here. You might not become an IT rock star overnight, but if you can do some basics with the command prompt, you'll find it's a short leap to create scripts.

Simple scripts are just a string of command prompt commands inserted into a batch file. If you can enter a command at the prompt, you can copy it and save it as a batch file, and you've created a script. The "Using Notepad to Create a Batch File" section in this chapter walks you through the steps to create a batch file.

There's a lot of depth to the command prompt that isn't apparent at first. However, as you dig into the commands, you'll see how much you can accomplish. This chapter isn't intended to make you an expert, but it will give you enough to get around and pass the related questions on the A+ exams. As your IT career progresses, you'll probably learn a lot more. If you want to view more details about any command prompt commands, check out this link: *http://technet.microsoft.com/library/cc772390.aspx*.

Starting the Command Prompt

The *command prompt* originated from the older Microsoft Disk Operating System (MS-DOS). Before Windows, all commands were typed at the command line. Windows has come a long way since the old MS-DOS days, but much of the same functionality of MS-DOS is built into the command prompt.

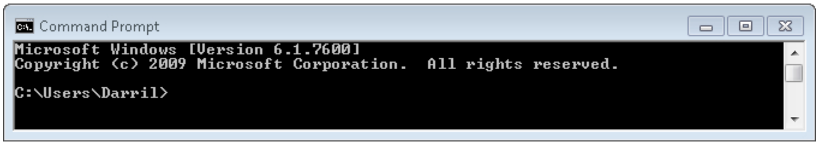
NOTE **CLI**

The command prompt is sometimes called the *command-line interface* (*CLI*).

Before you can start entering command prompt commands, you must first launch the command prompt. You can start it on Windows XP, Windows Vista, or Windows 7 by clicking Start, All Programs, Accessories, and selecting Command Prompt.

On Windows Vista and Windows 7, you can type **command** in the Search text box and select Command Prompt. On Windows XP, you can click Start, Run, type in **cmd**, and press Enter.

Figure 14-1 shows the basic Command Prompt window in Windows 7.



```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Darril>
```

FIGURE 14-1 Command Prompt window on Windows 7.

After the Command Prompt window opens, you can start entering commands. You type in a command, press the Enter key, and the command is executed. For example, you can type **ipconfig** at the Command Prompt window and press Enter to view Internet Protocol (IP) configuration information for your system.

The text preceding the greater-than character (>) is called the command prompt, or sometimes just the prompt. You'll see a blinking cursor after the >, prompting you to enter a command.

EXAM TIP

CompTIA has indicated that the 220-801 and 220-802 A+ exams will have performance-based questions. That is, instead of just multiple choice questions, you might have a simulation where you have to perform a task. The command prompt environment is a good candidate for these types of questions. Take the time to start the command prompt and enter all of the commands in this chapter, and you'll have a better chance at getting these performance-based questions correct.

Access Denied Errors

Occasionally, you'll need to start the command prompt with administrative privileges. For example, you might enter a command and see one of the following errors:

- Access is denied
- The requested operation requires elevation

These errors indicate that the command needs to be executed with administrative privileges. If you start the command prompt with administrative privileges, the command will execute without the error.

NOTE **ADMINISTRATIVE PRIVILEGES**

If you start a command prompt with administrative privileges, all commands entered within the prompt have administrative privileges. However, permissions in other applications are not affected.

Starting with Administrative Privileges

On Windows Vista or Windows 7, right-click Command Prompt and select Run As Administrator as shown in Figure 14-2. If prompted by User Account Control (UAC), click Yes to continue. On Windows XP, right-click Command Prompt, select Run As, and enter the credentials for an administrator account.

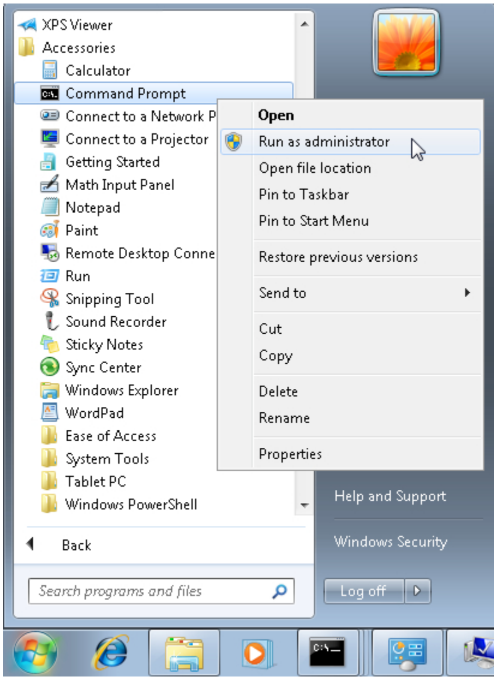
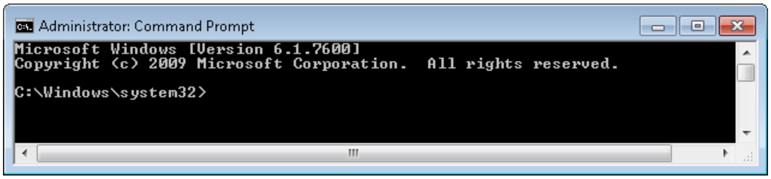


FIGURE 14-2 Launching the command prompt with administrative privileges.

Figure 14-3 shows the differences you'll see in the administrative Command Prompt window. You can see that the title bar is labeled Administrator: Command Prompt instead of just Command Prompt. The next section identifies some of the other differences.



```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>
```

FIGURE 14-3 Command Prompt with administrative privileges.

Command Prompt vs. Cmd

The application that runs the command prompt is cmd.exe. Another way of starting the command prompt in Windows Vista and Windows 7 is by clicking Start, typing **cmd** in the search text box, and selecting cmd. In contrast, the previous instructions said to type in **command** and select Command Prompt. There are some subtle differences.

If you use **cmd**, it will start the cmd.exe program directly, and you'll see C:\Windows \system32\cmd.exe in the title bar of the window. If you use **command**, it creates a shortcut in your user profile that points to the C:\Windows\system32\cmd.exe application. The same cmd.exe application runs, but the shortcut creates a slightly different environment, with Command Prompt in the title bar.

Click Start, type in **command**, right-click Command Prompt, and select Properties. This shows all the properties of the shortcut.

In contrast, if you click Start, type in **cmd**, right-click cmd, and select Properties, you'll see the properties of the cmd.exe application. There are fewer property options for the application than for the shortcut.

✓ Quick Check

1. **After running a command, you see an error indicating that the operation requires elevation. What should you do?**
2. **What is the name of the program that runs the command prompt?**

Quick Check Answers

1. **Start the command prompt with administrative privileges.**
2. **Cmd.exe.**

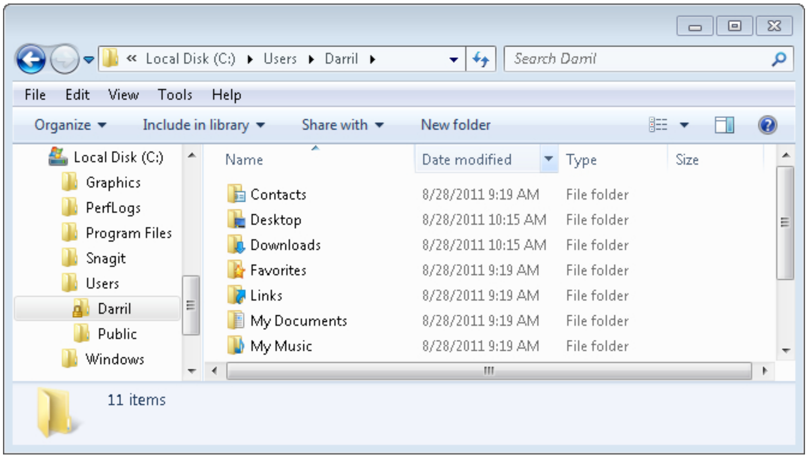
Understanding Paths

You might have noticed that the > prompt includes some other information. This identifies the path, which is the current disk drive, and the current *directory,* or *folder*. For example, in Figure 14-1, the path is C:\Users\Darril.

The system will use this as the default path for any commands that you execute. For example, if you enter the **dir** command to list the contents of the directory, it lists the contents of the current path. Figure 14-4 shows this same path from Windows Explorer.

NOTE FOLDERS VS. DIRECTORIES

You'll commonly hear the term *folders* to refer to the folders in Windows Explorer. These folders are the exact same folders you see from the command prompt, but when using the command prompt, you'll often hear the term *directories*. In other words, folders and directories are the same thing.


« Local Disk (C:) ▸ Users ▸ Darril ▸
Search Darril
File Edit View Tools Help
Organize ▾ Include in library ▾ Share with ▾ New folder
Local Disk (C:)
Graphics
PerfLogs
Program Files
Snagit
Users
Darril
Public
Windows
11 items

Name	Date modified	Type	Size
Contacts	8/28/2011 9:19 AM	File folder	
Desktop	8/28/2011 10:15 AM	File folder	
Downloads	8/28/2011 10:15 AM	File folder	
Favorites	8/28/2011 9:19 AM	File folder	
Links	8/28/2011 9:19 AM	File folder	
My Documents	8/28/2011 9:19 AM	File folder	
My Music	8/28/2011 9:19 AM	File folder	

FIGURE 14-4 Windows Explorer folders.

Default Paths

In addition to the current path you see from the command prompt, the system also knows about other paths. If you run a command that is not included in the current directory, the system will search through these other paths for the command.

For example, if you enter the ipconfig command, the system will first look in the current path (such as C:\Users\Darril) for ipconfig. If it can't find the command, it searches through other known paths.

You can view the known paths for your system by typing

at the prompt and pressing Enter. You'll see something like this:

```
PATH=C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\
WindowsPowerShell\v1.0\
```

Each path is separated by a semicolon, so this actually includes the following paths:

```
C:\Windows\system32
```

```
C:\Windows
```

```
C:\Windows\System32\Wbem
```

```
C:\Windows\System32\WindowsPowerShell\v1.0\
```

In this example, the ipconfig.exe program is located in C:\Windows\system32, so it will be run from that location. If the system can't locate the command in the current path or in any of the known paths, it will give an error. For example, if you incorrectly enter *ipconfig* as *ipnfig*, you'll see this:

```
'ipnfig' is not recognized as an internal or external command, operable program or batch
file.
```

Commands, Programs, and Batch Files

If you enter a valid command at the prompt, it will run. With this in mind, you might be asking yourself, "What is a valid command?" It can be one of the following:

- **Internal command**. The cmd.exe program includes many built-in commands. For example, the copy command is a core command and doesn't require an external application.
- **External command**. An external command is a command that is executed from an external file. It's not built into the cmd.exe program. For example, the ipconfig command will run the ipconfig.exe program located in the C:\Windows\system32 folder.
- **Executable program**. An executable program is any file that includes one of the following extensions: .com, .exe, .cmd, .vbs, .vbe, .js, .jse, .wsf, .wsh, and .msc. For example, dxdiag.exe is the program name for the DirectX Diagnostic Tool. If you enter dxdiag at the prompt, it will start the DirectX tool.
- **Batch file**. A batch file uses the .bat extension and is a grouping of one or more commands in a text file. When the batch file is executed, all of the commands are executed.

NOTE **EXTERNAL COMMANDS AND EXECUTABLE PROGRAMS**

An external command could also be considered an executable program because it usually has an extension of .exe. The major difference in this context is that external commands are considered those that have been around since the days of MS-DOS.

Understanding Extensions

In the early days of MS-DOS, all files were named with an eight-dot-three format. For example, ipconfig.exe includes eight letters (ipconfig), a dot, and a three-letter extension (.exe).

The extension is especially important. It tells the operating system what type of file it is, and the operating system associates many extensions with specific applications. For example, if someone emails you a file with .doc extension, you can double-click it and it will open Microsoft Word with the document showing. Similarly, if someone sends you a file with .xls extension, the file will open Microsoft Excel with the Excel spreadsheet showing.

NOTE **THE ASSOC COMMAND**

You can look at the known extension associations by entering assoc at the command prompt. You can view the applications that will start for specific extensions by entering ftype. Alternately, on Windows Vista and Windows 7, start the Control Panel, type assoc in the Search box, and select Change The File Type Associated With A File Extension to see the known extensions and associated applications.

When users were restricted to using only eight characters for the name, they often had to be creative when naming files. For example, an accountant might name a spreadsheet file stax1Q95.xls. It might not be apparent right away, but this is for state taxes for the first quarter of 1995.

Windows is no longer restricted to the eight-dot-three format. File names and extensions can now be longer, allowing you to name a file just about whatever you want. For example, the same accountant might name a file State Taxes First Quarter 2012.xlsx. If you double-click this file, it starts Microsoft Excel 2010 with the file showing.

Windows Vista and Windows 7 limit the full path, file name, and extension to 260 characters. For example, consider an Excel file stored in the C:\Data folder. The full name is C:\data \State Taxes First Quarter 2012.xlsx. This is 43 characters (C:\Data is 8 characters, and State Taxes First Quarter 2012.xlsx is 35 characters). Exceptionally long paths that cause the file to exceed the 260-character limit sometimes cause problems, especially when a user is trying to copy them.

✔ **Quick Check**

1. **What is the difference between a folder and a directory?**
2. **How can you identify paths known to your system?**

Quick Check Answers

1. **Nothing; they are the same.**
2. **By entering the path command and the command prompt.**

Command Prompt Basics

There are some basic rules and guidelines that are important to understand when using the command prompt. When you've grasped the basics, the more advanced topics are much easier to understand. The following sections cover basics from using uppercase and lowercase letters to using wildcards.

Case Sensitivity

For most commands entered within the command prompt, it doesn't matter what case you use when you enter them. You can use all uppercase, all lowercase, or a mixture. For example, the following three commands work exactly the same:

- **ipconfig**
- **IPCONFIG**
- **IPConfig**

There are some exceptions where you must enter the command using specific uppercase or lowercase letters, but they are rare. Where the case matters for any commands in this book, I'll stress its importance. Similarly, documentation usually provides emphasis if the case matters.

Understanding Switches

Most commands can be modified by using a switch. A switch is appended to the end of the command with a forward slash (/).

For example, the ipconfig command provides basic information on a computer's IP configuration. However, you can append it with the /all switch to get detailed information on the IP configuration. The command with the switch appended is ipconfig /all.

The switch should be preceded with a space. However, some commands will work without a switch. For example, ipconfig/all works the same as ipconfig /all. It's best to use a space before all switches because this will work consistently.

> ***NOTE*** **DASHES AND SLASHES**
>
> **Many commands also support the use of the dash (-) as a switch, instead of the slash (/). That is, you can use ipconfig -all and it will work the same as ipconfig /all.**

Different commands support different switches. Just because the /all switch works with ipconfig, you can't expect it to work with the xcopy command (as xcopy /all). If you don't know the valid switches for a command, the best thing to do is ask the system for help.

Getting Help

You can get help for almost all command prompt commands by appending the command with the help switch (/?). If you can't remember the exact syntax of the command, enter the command with /? to retrieve help on it.

Some commands also support the use of the help command. You can type in **help** followed by the command. The following two commands show how to get help on the chkdsk command:

```
chkdsk /?
```

```
help chkdsk
```

You can use these two commands for most commands by just substituting chkdsk with the name of the command. However, the help command isn't as universal as the /? switch. For example, if you enter **help ipconfig**, the system will instead prompt you to enter ipconfig /?.

EXAM TIP

When you can't remember the syntax of a command, use the /? switch to get help. This will show the syntax and all of the options that you can use with the command.

Using Quotes with Spaces

The command prompt interprets spaces as the next part of a command. For example, consider the copy command. The syntax is as follows:

```
copy sourceFile destinationFile
```

You could enter the following command to create a copy of the study.txt file and name the file copy backup.txt:

```
copy study.txt backup.txt
```

The command is copy and it recognizes the name after the space (study.txt) as the source file. It also recognizes backup.txt as the destination file because there is a space between study.txt and backup.txt.

However, file names can have spaces. What if the study.txt file was named A Plus Study Notes.txt? If you tried the following command, it wouldn't work:

```
copy A Plus Study Notes.txt Study backup.txt
```

It would see "A" as the name of the source file. However, you probably won't have a file named A, so it will return an error indicating that it can't find the specified file. The following command is a little better, but it will still fail:

```
copy "A Plus Study Notes.txt" Study backup.txt
```

In this case, it interprets everything within the quotes as the name of source file (A Plus Study Notes.txt). However, it then sees Study and backup.txt as two separate elements. It's expecting only one destination file, so it gives an error indicating the syntax of the command is incorrect. The correct command is:

```
copy "A Plus Study Notes.txt" "Study backup.txt"
```

Notice that both the source file name and the destination file name include spaces, so they are both enclosed with quotes.

NOTE USING QUOTES

Quotes aren't always needed, but using them is a good habit to get into. In some cases, certain characters aren't recognized, but enclosing the command in quotes resolves the problem.

Beware of Typos

There might come a day when computers will do what we want them to do instead of what we ask them to do. However, that day isn't here yet. With this in mind, you must type in the commands with the correct spelling.

For example, if you mean to enter ipconfig but enter ipcnfig instead (with the "o" missing), the system will give you an error indicating that it can't locate the program ipcnfig.

Sometimes, you'll see an error like the following:

```
The syntax of the command is incorrect.
```

This indicates that your computer understands the command name, but it can't recognize the other parts of the command. It might be that you're using invalid switches, missing a space, or not using quotes correctly.

When you see an error, one of the first things you should do is double-check the spelling. Often, you can find the problem by looking at the command one character at a time.

Recalling Commands

The command prompt keeps a history of all the commands you've entered in the current session. This is very useful if you need to reenter a command. You don't need to type it in from scratch every time.

Table 14-1 shows keys you can use to recall information in the command prompt history.

TABLE 14-1 Recalling Commands in History

Key	Comments
Up Arrow	Retrieves the previous command in the history list
Down Arrow	Retrieves the next command in the history list
Page Up	Retrieves the first command in the history list
Page Down	Retrieves the last command in the history list
Esc	Clears the current command

I strongly encourage you to play around with this. Launch a command prompt, enter a few commands, and then use these keys to see how they work. It can be a huge timesaver.

You can also use a few other keys to move the cursor around after you've retrieved a command. Table 14-2 shows some of these keys.

TABLE 14-2 Using Keys to Move the Cursor

Key	Comments
Home	Moves the cursor to the beginning of the command
End	Moves the cursor to the end of the command
Left arrow	Moves the cursor to the left one space
Right arrow	Moves the cursor to the right one space

For example, imagine that you entered the following command and the system complained with a syntax error:

```
copy "a+ study notes.txt" study backup.txt"
```

After looking at it, you realize you forgot the quote (") before Study. You could type it in from scratch, but you might actually end up with another typo. Instead, you could do the following:

1. Press the up arrow. This retrieves the last command.
2. Press the left arrow until your cursor is before the *S* in Study.
3. Type in the quote (**"**).
4. Press Enter. Note that you don't have to move your cursor to the end of the line to run the command.

NOTE **MOUSE DOES NOT CHANGE CURSOR POSITION**

It's common to try to reposition the cursor with the mouse, but it won't work in the Command Prompt window. You need to use the arrows to move the cursor's position.

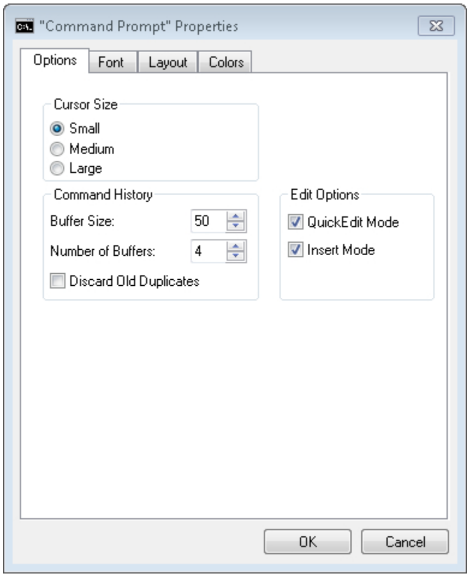
Copying and Pasting

You can also copy and paste data to and from the command prompt. It doesn't work with the typical Ctrl+C to copy or Ctrl+V to paste, but it is possible to copy and paste data into the command prompt. This can also save you some time.

Copying and pasting in the command prompt is easiest if you enable QuickEdit mode. Use the following steps to enable QuickEdit mode, and then copy and paste data to and from the Command Prompt window:

1. Launch the command prompt.
2. Right-click the title bar and select Properties.

3. Select the check box for QuickEdit mode. Your display will look similar to the following graphic. Click OK.



Cursor Size
● Small
○ Medium
○ Large

Command History
Buffer Size: 50
Number of Buffers: 4
□ Discard Old Duplicates

Edit Options
☑ QuickEdit Mode
☑ Insert Mode

***NOTE* PERSONAL PREFERENCES**

You can manipulate the appearance of the Command Prompt window by selecting the Font, Layout, and Colors tab. Many users like to manipulate the Colors tab for personal preferences.

4. Type **notepad** and press Enter. This will open an instance of the Notepad editor.
5. Type **ipconfig** within Notepad, but don't press the Enter key.
6. Press Ctrl+A to select the text you typed, and press Ctrl+C to copy the text to the clipboard.
7. Select the Command Prompt window.
8. Right-click within the Command Prompt window. The text you copied from Notepad is pasted into the window.
9. Press Enter to run the command.
10. Use the mouse to select the text displayed in the Command Prompt window. Click in the upper left of the Command Prompt window and drag the mouse to the bottom right. Release the mouse when you've completed the selection.

NOTE **SELECTING MULTIPLE PAGES OF TEXT**

You can also select multiple pages of text. For example, you can scroll to the beginning of the Command Prompt window and select from there. If you hold the mouse button down and drag the cursor below the bottom of the Command Prompt window, it will automatically scroll and select the text until you release the mouse button.

11. After you've selected the text, press Enter. You can also right-click the mouse. Both actions copy the text to the Clipboard.
12. Return to Notepad. Press Ctrl+V to paste the text into the Notepad document.

At this point, you can save the Notepad document. For example, you could use method to save the IP configuration of a system in a file.

Saving the Output in a Text File

The previous section showed how you can copy text from the command prompt, but sometimes you might want to send the output directly to a text file. You can do so with the > symbol, which redirects the output to a file.

For example, instead of copying the output of ipconfig /all and then pasting it into a text file, you could use the following command:

```
ipconfig /all > MyConfig.txt
```

After entering the command, you could open it with the following command:

```
notepad myconfig.txt.
```

You can also view the contents of the text file with the following command:

```
type myconfig.txt.
```

As an alternative, you can copy the output of a command to the Windows Clipboard using | clip, as follows:

```
ipconfig /all | clip
```

You can then paste the contents from the clipboard into any file with Ctrl+V.

Understanding Variables

Windows uses variables to identify many system elements. For example, every computer has a name, but the name will be different on each computer. Windows uses the %computername% variable to identify the name of the local computer.

All variables start and end with a percent symbol (%). You can view the value of a variable with the echo command. For example, you can view the name of your computer with the following command:

```
echo %computername%
```

Table 14-3 shows some common variable names along with what they represent. I encourage you to look at each of these names by opening a command prompt and typing in **echo** followed by the variable (for example, **echo %computername%**). If you enter a typo, the system will output what you entered with the percent symbols. In other words, if you see an output with the percent symbols, double-check your spelling. Press the up arrow to retrieve the previous command and correct it.

TABLE 14-3 Command Variable Names

Variable	Comments
%systemdrive%	System drive (such as C:\)
%systemroot% %windir%	Location of Windows (such as C:\Windows)
%computername%	Name of local computer
%homepath%	Path to user's profile (such as C:\Documents and Settings\username on Windows XP or C:\Users\username on Windows Vista and Windows 7)
%path%	List of paths known to the system
%temp%	Path to temporary folder in users profile
%username%	User name for logged on user
%userdomain%	Name of domain if computer is joined to a domain, or name of computer if computer is not joined to a domain
%userprofile%	Location of the profile for the logged on user

You can view a list of all variables known by your system by entering the **set** command. The output shows each variable name (without the percent symbols) along with the value of the variable. If you want to view the value of one or more variables, you can use the set command followed by the first letters of the variable. For example, **set system** and **set sys** both show the value of the system drive and system root variables.

NOTE **LOOKING AT VARIABLES IN THE GUI**

You can look at variables in Windows Vista or Windows 7 by clicking Start, typing Environment, selecting Edit The System Environment Variables, and clicking Environment Variables. Alternatively, you can click Start, right-click Computer, and select Properties, select Advanced System Settings, and select Environment Variables from the Advanced tab.

Using Wildcards

You can also use the following wildcards when using the command prompt:

- ***** The asterisk (*) can be used in place of zero or more characters.
- **?** The question mark (?) can be used in place of a single character.

For example, the dir command can be used to provide a listing of files in a directory. Table 14-4 shows you can use the * wildcard to view different results.

TABLE 14-4 Using the * Wildcard

Command	Result
dir	List all files in the current directory
dir d*	List all files that start with a d
dir *g	List all files that end with an g
dir *.txt	List all files with a .txt extension

The ? symbol can be used only for a single character. For example, imagine that you have several study documents but that each one has different versions, such as StudyV1.doc, StudyV2.doc, and so on. You could use the following command to list them all:

```
dir studyv?.doc
```

✔ **Quick Check**

1. **What are two methods of viewing help for the chkdsk command?**
2. **What can you press to retrieve the last command you entered at the command prompt?**

Quick Check Answers

1. **Chkdsk /? or help chkdsk.**
2. **The up arrow.**

File Commands

There are several commands you can use from the prompt to view and manipulate files and folders. This section identifies the relevant commands for the A+ exam and shows how to use them. While you're going through this section, I strongly encourage you to open up a command prompt and try these commands out.

Attrib

The *attrib* command is short for attribute, and it is used to list and manipulate attributes assigned to files and folders. Table 14-5 lists the attributes that can be assigned.

TABLE 14-5 Common File Attributes

Attribute	Comments
A Archive	When cleared, it indicates a file has been archived (or backed up). When set, it indicates the file has not been archived or has been modified since the last archive.
H Hidden	The file is not normally displayed unless the user changes settings to view hidden files.
I Not Index	Indicates whether the file is included in the index. Indexed files can be more easily located by the system.
R Read-only	Users cannot modify a read-only file without changing the attribute.
S System	The operating system protects system files to prevent users and malicious software from causing damage.

If you enter the attrib command at the command prompt, it will list all the files in the current directory along with any attributes that are assigned. For example, the following code listing shows the output from the attrib command. You can see that some of the files have specific attributes assigned.

Output from attrib command

```
C:\Study\A+Study>attrib
             C:\Study\A+Study\Archived File.rtf
A   H        C:\Study\A+Study\Hidden File.txt
A   I        C:\Study\A+Study\Not Indexed File.rtf
A   R        C:\Study\A+Study\Readonly File.rtf
A  SH        C:\Study\A+Study\System and Hidden File.sys
```

The first file in the output doesn't have any attributes showing. But others show A for not archived, H for hidden, I for not indexed, and S for System.

You can set or clear any attributes with the + or - symbols. For example, if you want to make a file named study.txt read only use the following command:

```
attrib +r study.txt
```

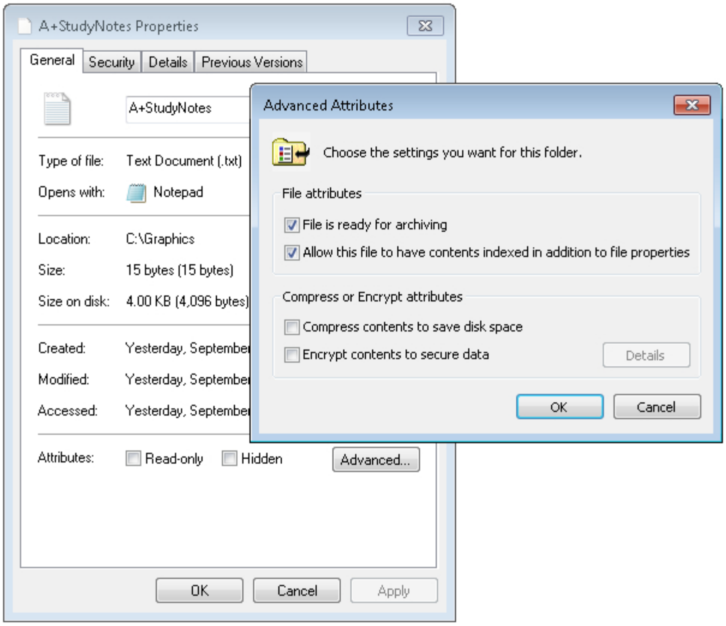
If you want to clear the read-only attribute, use the following command:

```
attrib -r study.txt
```

EXAM TIP

Sometimes you might need to delete or copy over a hidden, system, read-only file. This is not possible when these attributes are set. However, you can change the attributes by using the attrib command, and then the system will let you copy over the file.

You can also see most of these attributes from Windows Explorer. Right-click any file and select Properties. You can see the Read-only and Hidden attributes on the General tab of this page. Click the Advanced button, and your display will look similar to Figure 14-5. It's not common to manipulate the system attribute of a file, so you can't do so from Windows Explorer.


A+StudyNotes Properties

A+StudyNotes

Type of file: Text Document (.txt)

Opens with: Notepad

Location: C:\Graphics

Size: 15 bytes (15 bytes)

Size on disk: 4.00 KB (4,096 bytes)

Created: Yesterday, September

Modified: Yesterday, September

Accessed: Yesterday, September

Attributes: □ Read-only □ Hidden

Advanced Attributes

Choose the settings you want for this folder.

File attributes

☑ File is ready for archiving

☑ Allow this file to have contents indexed in addition to file properties

Compress or Encrypt attributes

□ Compress contents to save disk space

□ Encrypt contents to secure data

FIGURE 14-5 Attributes within Windows Explorer.

The compress and encryption attributes shown in Figure 14-5 can be manipulated with the compact and cipher commands, respectively. However, you're unlikely to come across these commands in the A+ exams.

Dir

The *dir* command is short for directory, and it is used to list files and directories. The "Using Wildcards" section earlier in the chapter showed some examples of how to use it. The following code listing shows the output you can expect from the dir command.

Using the dir command

```
C:\Study\A+Study>dir
 Volume in drive C has no label.
 Volume Serial Number is 78B1-B7A9
 Directory of C:\Study\A+Study
09/08/2012  02:16 PM    <DIR>          .
09/08/2012  02:16 PM    <DIR>          ..
09/08/2012  01:05 PM               198 A+Notes.rtf
09/08/2012  01:05 PM               198 Archived File.rtf
09/08/2012  02:12 PM                24 Hidden File.txt
```

```
09/08/2012  01:05 PM               198 Not Indexed File.rtf
09/08/2012  01:05 PM               198 Readonly File.rtf
               5 File(s)            816 bytes
               2 Dir(s)  119,378,903,040 bytes free
```

Notice that in addition to the files, the output also includes two other entries at the beginning: a single dot (.), and a double dot (..). The single dot refers to the root of the drive, which is C:\ in this case. The double dot refers to the parent folder, which is C:\Study in this case.

You can use the double dot to change to the parent directory as follows:

```
Cd ..
```

However, you can't use the single dot to change to the root. If you want to change to the root, you have to use the backslash, as follows:

```
Cd \
```

Some of the common switches used with the dir command are listed in Table 14-6.

TABLE 14-6 Common dir Switches

Switch	Comments
/w dir /w	Formats the output in a wide list with fewer details
/a dir /ah	Displays files that have specific attributes such as hidden (/ah), system (/as), or read only (ar)
/b dir /b	Displays files in a bare format without heading information or a summary
/q dir /q	Includes the name of the owner for each file
/s dir /s	Lists all files in the current directory, and all subdirectories

The dir command with the /s switch can be very useful when you're looking for a specific file. For example, imagine you're looking for a file that starts with study but you're not sure about the rest of the file's name. You can use the following command with wildcards:

```
dir study*.* /s
```

Or if you only know that the word study is somewhere in the name of the file, you can use the following command:

```
dir *study*.* /s
```

These commands will search subdirectories only within the current directory. If you want to search the entire drive from the root, you would need to either change the directory or include the path in the command. The following command would ensure that the search starts from the root of the drive by using the \ symbol to identify the root:

```
dir \*study*.* /s
```

Md

The *md* command is short for *make directory*, and it is used to create directories. Alternatively, you can use the mkdir command, which works exactly the same way. For example, if you wanted to create a directory named Study, you could use the following command:

```
md Study
```

You can use uppercase or lowercase for Study, and the directory will be created with the case you use. You can later access it by using either uppercase or lowercase.

You can also include the hard drive and the path, or just the path if you are creating the directory on the current drive. For example, if your current path is C:\Data and you want to create a directory at the root of the C: drive, you can use either of the following commands:

```
md C:\Study
```

```
md \Study
```

You can also create multiple directories with the same command. If your system didn't currently have a directory named Study but you wanted to create it along with a child directory named A+Study, you could use the following command:

```
md C:\Study\A+Study
```

The preceding command creates the Study directory and creates the A+Study directory as a child within C:\Study.

CD

The *cd* command is short for change directory, and it is used to change the current path. As an alternative, you can use the chdir command, which works exactly the same way.

As a reminder, all command prompt commands execute from the context of the current path. You can include the full path for commands or files in the command, but very often it's easier to change the directory.

For example, imagine that you are currently in the C:\Study\A+Study path, but you want to list the contents of the C:\Study folder. You could use the dir command with the path as follows:

```
dir C:\study
```

Or, you could first change the directory and then run the dir command as follows:

```
cd \study
dir
```

Notice that the backslash (\) is used in this command. The backslash refers to the root of the drive, so the command changes to the C:\Study directory.

Because you're only moving up one folder, you could also use the .. shortcut to change to the parent directory. If your current path is C:\Study\A+Study, the parent folder is C:\Study. The following commands can also work:

```
cd ..
dir
```

If you want to change to the root of the drive, you can use the following command:

```
cd \
```

Changing Drives

Often you'll want to view or manipulate files on different drives. For example, you might be in the C: drive but want to access files on the D: drive. To change drives, simply enter the drive letter followed by a colon, as follows:

```
C:\Windows\System32>d:

D:\>c:

C:\Windows\System32>
```

The first command (d:) changed to the D: drive, and the second command (c:) changed back to the C: drive.

Notice that when you change drives, the system uses a different path. The system remembers the path for each drive in the session, but if there isn't a path set, it uses the root. In the example, there wasn't a known path for D:, so it used the root of D:. However, it knew the path for C: was \Windows\System32, so it used this path when it returned to the C: drive.

Rd

The *rd* command is short for remove directory, and it is used to remove or delete directories. As an alternative, you can use the rmdir command which works exactly the same way.

Table 14-7 shows the two switches used with the rd command.

TABLE 14-7 Common rd Switches

Switch	Comments
/s Subdirectory	Used to remove a directory and all files and subdirectories within the subdirectory. This will prompt you with Are You Sure (Y/N). You can press Y to confirm the action or N to cancel it.
/q Quiet	Suppresses the Are You Sure question. This is useful if you include the command in a batch file.

For example, you can use the following command to delete all files and directories within a directory named test:

```
rd c:\test /s
```

EXAM TIP

If you try to remove a directory that has files or directories within it, you'll see an error unless you use the /s switch.

Del

If you want to delete individual files instead of full directories, you can use the *del* command. As an alternative, you can use the erase command, which works exactly the same way. The basic syntax is as follows:

```
del targetFile
```

You can use wildcards with the del command. For example, if you want to delete all the files in a directory, you can use the following:

```
del *.*
```

Table 14-8 shows some switches used with the del command along with examples.

TABLE 14-8 Common del Switches

Switch	Comments
/p Prompt	Prompts you for confirmation before deleting a file del study.txt /p
/f Force	Forces the deletion of read-only files del study.txt /f
/s Subdirectories	Deletes the file in all subdirectories del study.txt /s
/q Quiet	Suppresses confirmation prompt when using wildcards del *.* /q

Copy

As the name implies, the *copy* command can copy files and folders. The copy command was introduced in the "Using Quotes with Spaces" section earlier in this chapter. The basic syntax is as follows:

```
copy sourceFile DestinationFile
```

You can include the full path of the source file, the destination file, or both. It uses the current directory by default, so if either of the files are in different directories, you must include the directory for at least one.

For example, if your current path is C:\Study\A+ and you want to copy a file named A+Notes.txt to the C:\Study\Sec+ folder, you can use any of the following commands:

```
C:\Study\A+> copy "StudyNotes.txt" "C:\Study\Sec+"

C:\Study\A+> copy "StudyNotes.txt" C:\Study\Sec+\A+StudyNotes.txt"

C:\Study\A+> copy "C:\Study\A+\StudyNotes.txt" "C:\Study\Sec+"
```

NOTE **USING QUOTES**

These commands are using quotes to identify the source file and destination file. You don't always need quotes. However, in this example, the command prompt gets confused by the + symbol, so the quotes are needed for any paths that include the + symbol.

In the first example, the source file *path* isn't included, so the copy command looks for the StudyNotes.txt file in the current path (C:\Study\A+). In the second example, the name of the destination file is given, so the copied file will be named A+StudyNotes.txt. If the name isn't given, the same file name is used. The third example includes both the source path and the destination path.

You can also use wildcards with the copy command. The following commands provide two examples:

```
C:\Study\A+> copy *.* "C:\Study\Sec+"

C:\Study\A+> copy *.txt "C:\Study\Sec+"
```

The first command will copy all files in the current folder to the C:\Study\Sec+ folder. In the second example, it copies only files with the .txt extension.

Table 14-9 shows three important switches you should know about with the copy command.

TABLE 14-9 Common copy Switches

Switch	Comments
/a ASCII	Indicates files are ASCII-based text files. The copy command uses the end-of-file character (Ctrl+Z) with these files. If omitted, it considers the files binary files and doesn't look for the end-of-file character.
/v Verify	This verifies that the files are written correctly when they are copied.
/y Suppress prompts	If the same destination file already exists, the copy command prompts the user to confirm the operation before overwriting the destination file. The /y switch suppresses this prompt. This is useful if you include the command in a batch file.

A neat trick you can do with the copy command is copy the contents of multiple text files together. For example, imagine you have three log files named log1.txt, log2.txt, and log3.txt. You can use the following command to combine them all into a single file named logall.txt:

```
copy log1.txt + log2.txt + log3.txt logall.txt
```

Xcopy

The *xcopy* command is an extension of the copy command. This is a very rich tool, and all of its capabilities are beyond the scope of the A+ exam. However, the most important things to remember about the xcopy command are as follows:

- It can do everything that the copy command can do.
- It can copy directories and subdirectories.

As an example, if you wanted to copy the entire contents of the C:\Study directory, including all subdirectories, to the C:\Data directory, you could use the following commands:

```
xcopy c:\Study c:\Data\ /s

xcopy c:\Study c:\Data\ /s /e
```

The first command copies files and subdirectories but does not include subdirectories that don't have any files or directories within them. The second command includes empty subdirectories.

EXAM TIP

The xcopy command includes all of the same functionality of the copy command but adds a lot more. For example, the xcopy command can copy entire directories and subdirectories.

Robocopy

The *robocopy* command name sounds like it's related to robots, but it's actually short for *robust copy*. In this context, *robust* means strong and vigorous. It is commonly used to copy entire directories and includes all the features of copy and xcopy. While copy and xcopy will copy the files, they can't copy the file's metadata.

NOTE **METADATA (DATA ABOUT DATA)**

Metadata is information about data, or often stated as "data about data." For example, you might have a file named Studynotes.docx, and the content of the file is the data—your notes. The file's metadata includes information about the file, such as when you created it, when you last saved it, and who has permissions to access it.

When you copy a file, it creates a second file but it isn't a true copy. The data within the file will be the same, but some of the metadata will be different. Robocopy allows you to copy the file and all the metadata. Information that you can copy with robocopy includes the following:

- **Data**. This is the core information in the file, which will also be copied using copy and xcopy.
- **Attributes**. Both basic and advanced attributes can be copied.
- **Timestamps**. Original timestamps, such as when the file was created, are included in the copy.
- **Security information**. This includes all the permissions, such as the ability to read or modify a file.
- **Owner information**. Instead of identifying the person that performed the copy as the owner, the original owner information can be retained.
- **Auditing information**. Audit settings allow the system to log when someone accessed a file. Robocopy allows you to copy audit settings from the original file.

Table 14-10 shows some key switches you can use with robocopy.

TABLE 14-10 Common robocopy Switches

Switch	Comments and Example
/copyall	Copies the data and all metadata for files robocopy c:\notes d:\copies /copyall
/e	Includes all empty subdirectories robocopy c:\notes d:\copies /e
/purge	Deletes destination files and directories that no longer exist in the source location robocopy c:\notes d:\copies /purge
/move	Moves files and directories and deletes them from the source after they are copied robocopy c:\notes d:\copies /move

Sfc

The *sfc* command starts the System File Checker. This is a valuable tool that you can use to check the integrity of protected system files. This can be useful if a system has been infected with a virus. You can use the sfc tool to detect whether the virus has tried to modify any system files.

Sfc can detect and repair problems with system files. Table 14-11 shows the common switches you can use with sfc.

TABLE 14-11 Common sfc Switches

Switch	Comments
/scannow	Scans all protected system files. It will attempt to repair files and will report on its success or failure. You can view a report from the scan by looking here: %windir%\logs\cbs\cbs.log. The command is: sfc /scannow
/verifyonly	This runs a scan, but it does not attempt to fix any problems. The command is: sfc /verifyonly
/scanfile	Instead of scanning all protected system files, it scans only the specified file. For example, you can use the following command to scan a specific file: sfc /scanfile=c:\windows\system32\kernel32.dll
/verifyfile	This runs a scan on the specified file, but it does not attempt to fix any problems. Here's an example: sfc /verifyfile=c:\windows\system32\kernel32.dll
/purgecache	This switch is available only in Windows XP. It removes the contents of a folder named %windir%\system32\dllcache and repopulates the folder with verified files.

Using Notepad to Create a Batch File

Notepad is a text editor accessible from the command prompt. You can use the following steps to open Notepad and create a batch file.

1. Launch the command prompt.
2. Type in **notepad** and press Enter. This will open the Notepad editor.
3. Within the editor, type in **ipconfig /all > c:\scripts\MyConfig.txt**. Note that if you entered this at the command prompt, it would send the output of the ipconfig /all command and save it into a file named myconfig.txt, in a folder named MyConfig.txt.
4. Press Alt+F. This allows you to access the File menu.
5. Use the down arrow to select Save As. Browse to the root of C:, and click New Folder.
6. Name the folder **Scripts**. Double-click the Scripts folder to open it.
7. Type in **ipc.bat** as the name of the file. Press Enter to select Save. At this point, you have created a batch file named ipc.bat in the c:\scripts folder, with the ipconfig /all command within it.
8. Press Alt+F to access the File menu again.
9. Use the down arrow to select Exit, and press Enter. You'll be returned to the command prompt.
10. Type in **c:\scripts\ipc.bat** to run the batch file you just created. This will run the ipconfig /all command and redirect the output to a text file named MyConfig.txt.
11. Type in **Notepad c:\scripts\MyConfig.txt**. This will open the text file you just created with the Notepad editor.

Operating System Commands

There are many commands that you can enter at the command prompt that open a Windows-based application. You can often get to the same application from within the Start menu. However, the name used to launch these applications has become well known by many technicians and provides an easy shortcut to them.

EXAM TIP

It's valuable to know the name of the command used to open these tools and what each of the tools can do.

Table 14-12 lists some common tools that you can launch by entering the appropriate command at the command prompt, along with the chapter where the tool is covered in more depth. Many of the commands can also be launched from the search text box in Windows or Windows Vista, or from the Run line in Windows XP.

TABLE 14-12 Common Operating System Commands

Command	Tool Description	Chapter
dxdiag	DirectX Diagnostic Tool. Used to troubleshoot multimedia issues related to games or movies using DirectX.	6
explorer	Starts Windows Explorer. Note that Windows Explorer is used to view files on a system and is different from Internet Explorer, which is used to browse the Internet.	13
mmc	Opens a blank Microsoft Management Console. You can add snap-ins to an MMC.	13
msconfig	System Configuration tool. Used to configure the system, services, and startup applications.	15
msinfo32	System Information tool. Identifies hardware resources, components, and the software environment. This is useful for identifying the BIOS version, the amount of RAM installed on a system, and the processor type and speed.	2,3
mstsc	Opens the Remote Desktop Connection window, which can be used to connect to another system.	20
regedit regedt32	Opens the registry editor.	17
Services.msc	Opens the services applet. This can be used to stop, start, enable, and disable services.	13
sigverif	The File Signature Verification utility is used to identify digitally signed files and verify their integrity.	15

NOTE **RUN LINE**

You can access the Run line in Windows Vista and Windows 7 by pressing the Windows key and the R key at the same time. The Windows key is typically between the Ctrl and Alt keys to the left of the spacebar.

I strongly encourage you to start each of the programs listed in Table 14-12 and look at them. You don't need to master them at this point, but take a look around and see what information is available. If something interests you, look deeper. You'll be a step closer to mastering them for the A+ exams, and you'll have a better idea of what tool to use when you need to troubleshoot a computer.

On Windows Vista and Windows 7, click Start, type the name of the program in the Search text box, and press Enter.

On Windows XP, click Start, Run, type the command, and press Enter.

Table 14-13 lists some miscellaneous tools you can launch from the command prompt.

TABLE 14-13 Miscellaneous Operating System Commands

Command	Tool Description
date	View or set the date.
kill	Used on Unix-based operating systems to terminate a process.
shutdown	You can use this to force a shutdown (/s), restart (/r), or logoff (/l) from the Command Prompt.
time	View or set the time.
tlist *tasklist*	Shows a list of running processes. On Windows Vista and Windows 7, tlist has been renamed to tasklist.
type	Sends the output of a text file to the screen.
ver	Shows version information for the operating system.
winver	Opens the About Windows page, showing version information.

Disk Commands

Windows includes several commands that you can use to view, manipulate, and modify disks. These are described in detail in Chapter 16, "Understanding Disks and File Systems," but for an overview, they are listed in Table 14-14. These are entered at the Command Prompt, and each of them has help available by entering the command with the /? switch.

TABLE 14-14 Common Disk Commands

Command	Description
chkdsk	Checks and fixes disk errors.
convert	Converts a file system from FAT or FAT32 to NTFS. Data on the disk is retained.
format	Formats a partition with a file system. All data on the disk is lost.
diskpart	Tool used to manipulate hard disks. Diskpart replaces the older fdisk tool.
defrag	Defragments a volume for better performance.
fdisk	Legacy tool used to format and partition hard drives (replaced by diskpart).

EXAM TIP

It's best to run the disk commands in Table 14-14 and the networking commands listed in Table 14-15 at the command prompt. You should not run them from the Run line in Windows XP or the Start, Search text box in Windows Vista and Windows 7, because the Command Prompt window will open, run the command, and then close. You might not be able to see the results of the command. Try it with the ipconfig command.

Networking Commands

There are several common commands you can use to manipulate and troubleshoot a network. These commands are explored in greater depth in later chapters. However, as an introduction, they are listed in Table 14-15 with a short description and the chapter where they are covered.

TABLE 14-15 Common Networking Commands

Command	Description	Chapter
ipconfig	Displays IP configuration for network interface cards	24
ping	Checks connectivity with a device on a network	24
tracert	Checks connectivity with a device and shows the path through routers	24
nslookup	Used to query Domain Name System (DNS) Can verify records exist in DNS to resolve host names to IP addresses	24
arp	Shows mapping of IP addresses to media access control (MAC) addresses	24
nbtstat	Shows statistics for connections using NetBIOS over TCP/IP	24
netstat	Shows inbound and outbound connections	24

netsh	Advanced command used to manipulate and show network settings	22
net	Group of commands to view and manipulate network settings	24
telnet	Used to connect to remote system from the Command Prompt	20

EXAM TIP

When troubleshooting connectivity issues, ping and ipconfig are two of the most commonly used tools. Other commands are valuable, but you can often identify a problem by first checking the IP configuration with ipconfig and then checking connectivity with ping.

Chapter Summary

- The command prompt is a text-based window where you can enter MS-DOS–based commands.
- You can start the command prompt by entering **cmd** at the Run line in Windows XP. In Windows Vista or Windows 7, click Start, type **command** in the Search text box, and press Enter.
- If a command gives an access denied or command needs elevation error, start the command prompt with administrative privileges.
- You can get help on most commands by entering the command followed by the /? switch (such as **ipconfig /?**), or by entering **help** followed by the command (such as **help ipconfig**).
- The attrib command is used to view and manipulate attributes.
- Commands related to directories are dir to show the contents, md to make directories, cd to change directories, and rd to delete directories. The del command can delete individual files.
- Copy is used to copy files, and xcopy is an extended version of copy. Xcopy can do everything that copy can do and can copy subdirectories. Robocopy can do everything that copy and xcopy can do and can also copy metadata.
- Sfc (System File Checker) is used to verify the integrity of system files. The /purgecache switch included in Windows XP is not available in Windows Vista and Windows 7.
- Many applications can be launched from the command prompt, the Run line in Windows XP, or the Search text box in Windows Vista or Windows 7. You just need to know the command. Some common commands are:
  - Dxdiag starts the DirectX Diagnostic Tool.
  - Msconfig starts the System Configuration tool.
  - Msinfo32 starts the System Information tool.

  - Mstsc opens the Remote Desktop Connection window.
  - Regedit and Regedt32 open the registry editor.
  - Sigverif starts the File Signature Verification tool.
  - Shutdown can be used to shut down or log off of a system.
  - Tlist or tasklist can be used to show running processes.
- Some of the common disk commands that you can execute from the command prompt are:
  - Chkdsk can check and fix disk errors.
  - Convert can change a FAT or FAT32 file system to NTFS.
  - Format is used to format a partition with a file system such as NTFS.
  - Diskpart can view and manipulate hard disks and partitions.
  - Defrag can defragment a volume for better performance.
  - Fdisk is an older tool used to format and partition hard drives, but it has been replaced by diskpart.
- Some of the common networking commands that you can execute from the command prompt are:
  - Ipconfig displays IP configuration for network interface cards.
  - Ping checks connectivity with a device on a network.
  - Tracert checks connectivity with a device and shows path through routers to the device.
  - Nslookup checks Domain Name System (DNS) for name resolution records.
  - Netstat shows inbound and outbound connections for a computer.
  - Netsh is an advanced command used to manipulate and show network settings.
  - Net is a group of commands used to view and manipulate network settings.
  - Telnet is used to connect to remote system from the Command Prompt.

Chapter Review

Use the following questions to test your knowledge of the information in this chapter. The answers to these questions, and the explanations of why each answer choice is correct or incorrect, are located in the “Answers” section at the end of this chapter.

1. You entered the chkdsk command at the command prompt but received a syntax error. What command can you enter to show you the proper format of the command? (Choose all that apply.)

   A. chkdsk /syntax

   B. chkdsk /?

C. help chkdsk

D. chkdsk /show

2. You need to copy over a system file in Windows XP. What command can you use to change the system attribute?

   A. msinfo32

   B. msconfig

   C. regedit

   D. attrib

3. Which of the following commands will change a file named Study.doc to a read-only file?

   A. attrib +r study.doc

   B. attrib +ro study.doc

   C. attrib +h study.doc

   D. attrib +s study.doc

4. What command can you use to delete a folder from the command prompt?

   A. md

   B. cd

   C. rd

   D. df

5. What can you do with the copy command that you can't do with the xcopy command?

   A. Nothing. The xcopy command can do everything that the copy command can do.

   B. You can copy entire subdirectories.

   C. You can suppress prompts.

   D. You can use wildcards.

6. A Windows 7 system was recently infected with a virus. After removing the virus, you want to verify the integrity of system files and repair any problems. What command would you use?

   A. sfc /purgecache

   B. sfc /scannow

   C. sfc /verifyonly

   D. msconfig

7. You suspect that the kernel32.dll file is corrupt. Which command can you use to verify the integrity of the file?

   A. chkdsk /verifyfile=c:\windows\system32\kernel32.dll

   B. sfc /verifyfile=c:\windows\system32\kernel32.dll

   C. attrib /verifyfile=c:\windows\system32\kernel32.dll

   D. cd /verifyfile=c:\windows\system32\kernel32.dll

Answers

1. **Correct Answer:** B, C
   - **A. Incorrect:** The /syntax switch is not a valid switch.
   - **B. Correct:** Almost every command supports the /? switch as a method to retrieve help.
   - **C. Correct:** Most commands support the help command followed by the name of the command as a method to retrieve help.
   - **D. Incorrect:** The /show switch is not a valid switch.
2. **Correct Answer:** D
   - **A. Incorrect:** You can start the System Information tool with msinfo32 and view hardware resources, components, and the software environment.
   - **B. Incorrect:** You can start the System Configuration tool with msconfig to configure the system, services, and startup applications.
   - **C. Incorrect:** You can open the registry editor with regedit.
   - **D. Correct:** The attrib command allows you to modify attributes of files.
3. **Correct Answer:** A
   - **A. Correct:** The +r switch of the attrib commands set the read-only attribute of a file.
   - **B. Incorrect:** The attrib +ro command is not valid.
   - **C. Incorrect:** The attrib +h command sets the hidden attribute of the file.
   - **D. Incorrect:** The attrib +s command sets the system attribute of the file.
4. **Correct Answer:** C
   - **A. Incorrect:** The md command makes a directory.
   - **B. Incorrect:** The cd command changes the path to a different directory.
   - **C. Correct:** The rd command removes a directory (which is also known as a folder).
   - **D. Incorrect:** The df command isn’t valid and will fail.
5. **Correct Answer:** A
   - **A. Correct:** The xcopy command can do everything that the copy command can do, plus a lot more.
   - **B. Incorrect:** You can copy subdirectories with the xcopy command but not with the copy command.
   - **C. Incorrect:** Both commands allow you to suppress the prompts.
   - **D. Incorrect:** Both commands support the use of wildcards.

6. **Correct Answer:** B

   A. **Incorrect:** The /purgecache switch worked in Windows XP but does not work in Windows Vista and Windows 7.

   B. **Correct:** The sfc /scannow command will scan all protected system files and attempt to repair them.

   C. **Incorrect:** The sfc /verifyonly command will scan all protected system files, but it does not attempt any repairs.

   D. **Incorrect:** The msconfig command starts the System Configuration tool, which can be used to configure the systems, services, and startup applications.

7. **Correct Answer:** B

   A. **Incorrect:** The chkdsk command can check and fix a disk, but the given command will fail with a syntax error.

   B. **Correct:** This is a valid command to check the file.

   C. **Incorrect:** The attrib command shows attributes, but the given command will fail with a syntax error.

   D. **Incorrect:** The cd command changes the directory, but the given command will fail with a syntax error.