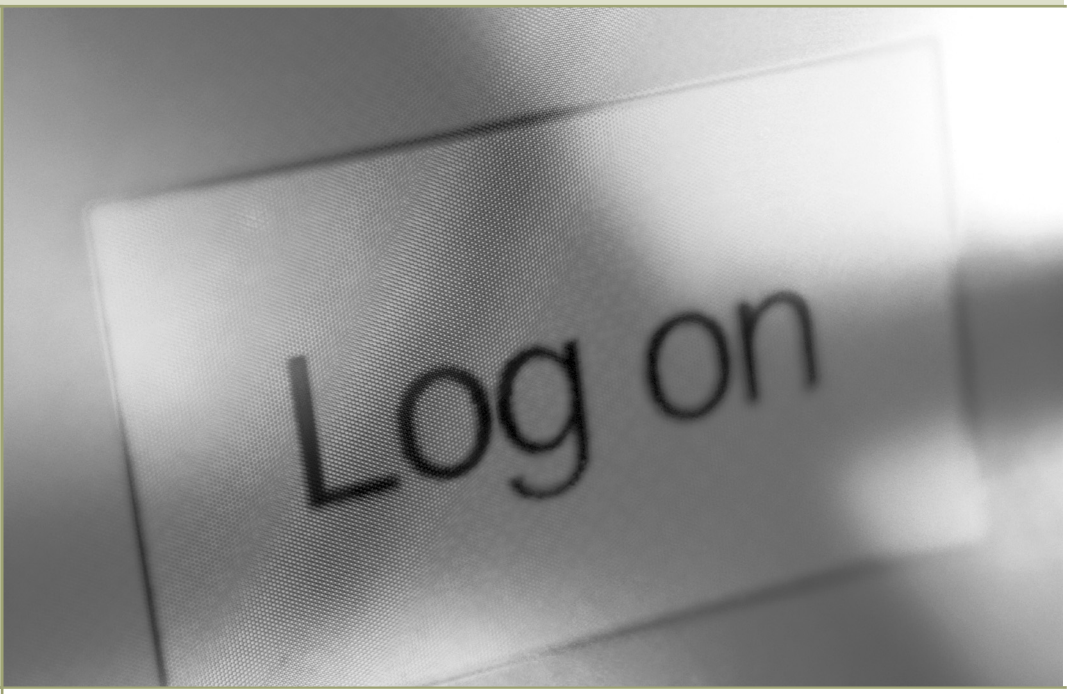
chapter 20

Zen and the Art of Network Support

"*Primum non nocere*"
(*"First, do no harm."*)

—Attributed to Galen of Pergamum (131–201 A.D.), Greek physician



In this chapter, you will learn how to

- **Describe troubleshooting tools**
- **Explain the troubleshooting process**
- **Analyze troubleshooting scenarios**

Have you ever seen a tech who walks up to a network and seems to know all the answers, effortlessly typing in a few commands and magically making the system or network work? I've always been intrigued by how they do this. Observing such techs over the years, I've noticed that they tend to follow the same steps for similar problems—looking in the same places, typing the same commands, and so on. When someone performs a task the same way every time, I figure they're probably following a plan. They understand what tools they have to work with, and they know where to start and what to do second and third and fourth until they find the problem. This chapter's lofty goal is to consolidate my observations on how these "übertechs" fix networks. We'll look at the primary troubleshooting tools, formulate a troubleshooting plan, and learn where to look for different sorts of problems. At the end of the chapter, we'll apply this knowledge to some common troubleshooting scenarios.

Test Specific

■ Troubleshooting Tools

While working through the process of finding the cause of a problem, you will need to use many tools. Some of these tools are difficult to quantify—including things like asking questions, referring to your network baselines and documentation, and synthesizing your network knowledge. Other tools are easier to describe; these are the software and hardware tools that provide information about your network. Many of the tools that fall into this category have been described already, such as hardware or software loopback testing devices, utilities like PING and TRACERT, and hardware tools like tone locators. The trick is knowing when and how to use these tools to solve your network problems.

"Touchy" Tools

The tools that are the most difficult to quantify, because they are mostly within you, are what I call **touchy tools**. An example of a touchy tool is asking questions of the person who has the problem. This is touchy because there's no predetermined set of questions to ask—your background knowledge and intuition must tell you which questions are the right ones. The *types* of questions you should ask can be grouped into a few categories:

- *Questions designed to find out exactly what steps the user took that may have caused the symptom.* This information can help you re-create the problem. In addition, these types of questions can provide insight into whether the problem was caused by user error or improper procedures. Note from the Geek Central Human Relations Department: be careful how you phrase these sorts of questions. An annoyed, "What did you do this time, you pinhead?!" tends to cause people to clam up. Your goal, remember, is to extract information.
- *Questions designed to find out exactly what any error messages said and the context of the error.* This information can be critical when you need to search manufacturers' knowledge bases and support lines.
- *Questions designed to find out what the user has tried to do to fix the problem.* This information can help you determine whether you are dealing with multiple layers of problems. Many users will try to fix things, but won't think to write down what they're doing. When at some point in the process they reach a standstill and can't get back to where they started, they will come crying to you for help. Be gentle! Your goal is to get the user to remember most, if not all, of the steps they tried, so you can backtrack to the original problem rather than troubleshooting a multilayered one.

No matter what the problem, always consider the safety of your data first. Ask yourself this question *before* you perform any troubleshooting action: "Is what I'm about to do going to potentially damage data?"

Another touchy tool you can use is to compare the current situation to the baselines and documentation you created for your network. When users complain of slow connections or downloads, for example, you should compare the bandwidth and connection speeds from your baseline to what you are able to test while the problem exists. You may find that the difficulty lies more with the user's expectations than any real network problem. You can then decide whether to upgrade your systems and connectivity, or explain to your users the limitations of the network.

The most complicated touchy tool to describe and quantify is the network knowledge you have that you can apply to your network's problems. The Network+ exam contains many network troubleshooting questions that you should be able to answer, not from reading this chapter, but from reading the rest of the book. Troubleshooting often comes down to applying prior knowledge in a new way. You know, for example, that an IP address, a subnet mask, and a default gateway are all necessary if your network is to communicate with the Internet using the TCP/IP protocol. Edgar complains that he cannot connect his Windows XP client to the Internet. His hardware seems to be functioning correctly (link lights are on and he connects to the server) and the proxy server is up. At that point, you might check his TCP/IP configuration by using the IPCONFIG utility. If you notice that he has no default gateway, you have solved the problem. The knowledge to solve this problem came partially from understanding how to troubleshoot by eliminating possibilities, but it also came from your knowledge of the elements required to connect a machine to the Internet. As you prepare for the exam, and for administering your company's network, ask yourself how each thing you learn about networking could be applied toward troubleshooting the network. This prepares you for the time when you have to make that leap.

Try This!

Roleplay

You can try out the touchy skills with a partner (classmate, friend, family member—any of these will do nicely) pretty easily by roleplaying. Ask your partner to come up with a computer- or technology-related problem, and then ask them questions. Try to focus your questions by using the touchy skills. Don't accuse your partner of breaking whatever, for example, but rather figure out whether it worked in the past or not. Run through several sessions of roleplay with the same partner so you can get a good sense of how to create scenarios (my PC is dead, I can't access the network, and so on) that are fully developed.

Hardware Tools

In Chapter 8, "Installing a Physical Network," you read about a few hardware tools you use when configuring your network. These **hardware tools** include cable testers, protocol analyzers, hardware loopback devices, and toners. These tools can also be used in **troubleshooting scenarios** to help you eliminate or narrow down the possible causes of certain problems. In addition, there are other pieces of hardware which, although they can't actively be used for troubleshooting, can provide clues to the problems you face.

A cable tester enables you to determine if a particular cable is bad. *Bad* can be defined in a variety of ways, but it essentially means that the cable is not delivering the data for whatever reason—perhaps the cable is broken, crimped badly, or sits too close to a heat or electrical source. In most troubleshooting

Cross Check

Connecting to the Internet

This chapter assumes you have the tools needed by PCs to connect to the Internet pretty much nailed down in your mind, so now is a good time to cross-check your memory! Turn to Chapter 11, "TCP/IP," and see if you can answer these questions. What's the purpose of the default gateway? How does this gateway manifest on a network? Why do you need an IP address? Wouldn't a MAC address suffice? And what's the point of a subnet mask?

situations, you will use other clues to determine if you have a hardware or software problem. Then, if you have narrowed down the problem to a hardware connectivity issue, a cable tester can help you determine if the cable is good or bad. Another option, if you are without one of these tools, is to replace the cables (one at a time) and test the connectivity. In the "I can't log on" scenario, for example, if you have determined that everyone else in the area can log on, and that this user can log on from another location, you have narrowed the problem to either a configuration or hardware issue. If all network activity is broken (in other words, nothing is available in Network Neighborhood or you cannot ping the default gateway), you may choose to test cables connecting the PC to the server. This is not the only option, but it is one variable that can be tested and eliminated.

Protocol analyzers come in both hardware and software flavors. Most of the hardware analyzers have an additional software component. These tools enable administrators to determine what types of traffic are flowing through their networks. Most analyzers translate the packets flowing over the network to provide destination, source, protocol, and some information about content. In a situation where you have a network slowdown, you might use a protocol analyzer to determine what types of packets are passing over your network, and where they are originating. In some cases, a dying network card can produce large numbers of spurious packets, which can be detected by using a protocol analyzer and noticing that all of the packets are coming from one location. Once you have narrowed your problem down to a particular machine, you can concentrate on that rather than blaming your servers, bandwidth, or other elements.

Software Tools

Throughout the book, you have read about **software tools** for configuring your network that can also be applied to troubleshooting. Because most of these have been described in previous chapters, I'll just review the basic purposes of these tools here. Key software **troubleshooting tools** include the following:

- **Software-Based Protocol or Network Analyzers** These include applications such as the **Network Monitor (NetMon)** provided with most versions of Windows. Also called *packet sniffers*, such as the popular Ethereal, included with the CD in this book, **network analyzers** collect and analyze individual packets on a network to determine bottlenecks or security breaches. Use these tools when you have unexplained slowdowns on your network, to help determine which machines are sending packets. This enables you to determine if there is too much broadcasting in general, or if you are the victim of some more sinister event like a hacker attack.
- **System Logs** Applications like Windows NT/2000/XP/2003's Event Viewer create **system logs** that display any errors or problems that have occurred in your system. If a user repeatedly fails at his logon, for example, this might be recorded in the appropriate view in the Event Viewer tool. That information could be the clue you need to determine that the user is locked out, either because he forgot his password or because someone has been trying to hack into that

account. Logs also provide information on services or components of the operating system that won't start or are receiving errors. This is a good way to troubleshoot system crashes on a particular machine.

- **Performance Monitors** Tools like the Performance Monitor mentioned earlier in this chapter can provide clues to the utilization pattern of a particular machine. When users complain of slowdowns or logon problems that can be traced to a specific system or server, this tool can often give you clues as to what's happening on that system that could be causing problems. For example, if a system is going to 100 percent memory utilization when a particular application is started, it may mean that you need more RAM, or perhaps that you need to put that application on a dedicated server. Use this tool for troubleshooting when you have tracked the problem to a particular machine and now must determine where in that machine the bottleneck exists.

Your Toolbox

It always amazes me when a person calls and asks me what tools they need to become a network tech. My answer is always the same: just your brain—everything else will pretty much appear when you need it. There is no such thing as the correct network tech's toolbox, full of hardware and software tools you haul around to get networks fixed. I certainly don't haul around cable testers, toners, or Time Domain Reflectometers (TDRs). I may bring them along if I suspect a cabling problem, but I normally won't dump them in my backpack until *after* I suspect a problem.

Software tools all come with the network operating systems themselves. I don't need a floppy disk with PING, for example, because it's on every PC in the house! The software tools you need to fix a network are sitting there, ready for you to use. Your task is to know *how* to use them. It's the *know-how* you need to bring, and in most cases, little or nothing else!

My lack of dependence on a toolkit makes some customers unhappy—they just can't reconcile in their minds that I can get their networks up and running without some big toolbox. At times this has become so much of an issue that I have brought along a toolkit just to look good! Seriously! I call that box full of stuff my prop because it's of no more real use than a rubber stage sword.

Now that we're no longer obsessing on carrying the right tools, let's instead concentrate on the real toolbox—your brain. In an amazingly brazen act of self-confidence, I have taken the liberty of quantifying the many mental processes you should use when fixing networks into the following section.

■ The Troubleshooting Process

Troubleshooting is a dynamic, fluid process that requires you to make snap judgments and act on them to try and make the network go. Any attempt to cover every possible scenario would be futile at best, and probably also not in your best interests, because any reference that tried to list every troubleshooting problem would be obsolete the moment it was created. If an exhaustive listing of all network problems is impossible, then how do you decide what to do and in what order?

Before you touch a single console or cable, you should remember two basic rules: as the Greek/Roman physician Galen wisely admonished, "First, do no harm." If at all possible, don't make a network problem bigger than it was originally. This is a rule I've broken thousands of times, and you will too. But if we change the good doctor's phrase a bit, it's possible to formulate a rule you can live with: "First, do not trash the data!" My gosh, if I had a dollar for every megabyte of irreplaceable data I've destroyed, I'd be rich! I've learned my lesson, and you should learn from my mistakes: Always make good backups! The second rule is to create **baselines** (that is, data on how the network runs when nothing is broken). **Backups** enable you to rebuild a trashed system. Baselines enable you to compare your network's current performance to how it behaves when all is well. Both of these steps should precede *any* actual troubleshooting activity on your part.

Once you've done your backups and baselines, and it's time to start typing commands or yanking cable, you need a plan. You need to know what steps to take and in what order. These steps are pretty much universal to all network problems, and you should learn them well. You also need a guide to where you should look in the network for problems. For that job, I will gift you with what I modestly call **Mike's Four-Layer Model**. As in troubleshooting, so in this chapter—first, backups and baselines, and then you can start taking troubleshooting steps with my model to guide you.

Backups

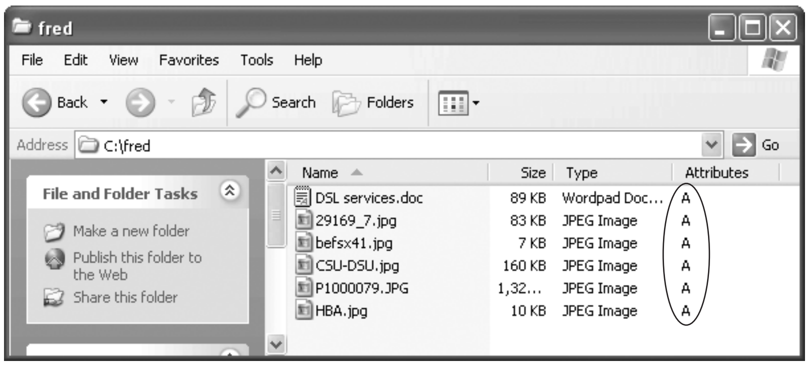
Think about how much work you have done to create nice, stable servers, responsive workstations, and overall network wonderfulness. Imagine how many hours your users have spent creating data and storing it on those servers. Now imagine a virus deleting critical data or configuration files—not a good situation, for either your blood pressure or your job security. Simple common sense dictates that you create backups of your data. Repeatedly. Backups are useless unless they are current. Your users won't thank you for restoring a three-week-old copy of a file that gets updated daily. You should therefore create a backup schedule for your data that ensures it's backed up often enough that a useful copy can be restored easily. Your backup plan should include the following details:

- When the backups will occur and what the tape rotation schedule will be
- What types of backups will be done at each time
- Where the backups will be stored

Perhaps the most important details are the types of backups and the schedule, or strategy, for the backups.

What Are the Types of Backups?

The goal of backing up data is to ensure that when a system dies, there will be an available, recent copy you can use to restore the system. You could simply back up the complete system at the end of each day—or whatever interval you feel is prudent to keep the backups fresh—but complete backups



C:\fred

Name	Size	Type	Attributes
DSL services.doc	89 KB	Wordpad Doc...	A
29169_7.jpg	83 KB	JPEG Image	A
befsx41.jpg	7 KB	JPEG Image	A
CSU-DSU.jpg	160 KB	JPEG Image	A
P1000079.JPG	1,32...	JPEG Image	A
HBA.jpg	10 KB	JPEG Image	A

• **Figure 20.1** The archive bit

can be a tremendous waste of time and materials. Instead of backing up the entire system, take advantage of the fact that all the files won't be changed in any given period; much of the time you only need to back up what's changed since your last backup. Recognizing this, most backup software solutions have a series of options available beyond the old complete (usually called Full or Normal) backup.

The key to understanding backups other than the full backup is a little fellow called the Archive attribute. All files have little 1-bit storage areas called *attributes*. The most common attributes are Hidden (don't show the file in My Computer or when **DIR** is typed at the command line), System (it's a critical file for the system), Read-Only (can't erase it), and the archive bit. These attributes were first used in FAT-formatted drives in the DOS era, but they are still completely supported today by all file formats. The **archive bit** works basically like this: whenever a file is saved, the archive bit is turned on. Simply opening a file will affect the current state of the archive bit. Backup programs will usually turn off a file's archive bit when it's backed up. In theory, if a file's archive bit is turned off, it means there's a good backup of that file on some tape. If the archive bit is turned on, it means that the file has been changed since it was last backed up (see Figure 20.1).

Archive bits are used to perform backups that are not full backups. The following backup types are most often supported:

- A **normal backup** is a full backup. Every file selected will be backed up, and the archive bit will be turned off for every file backed up. This is the standard "back it all up" option.
- A **copy backup** is identical to a normal backup, with the important distinction that the archive bits are *not* changed. This is used (although not often) for making extra copies of a previously completed backup.
- An **incremental backup** includes only files with the archive bit turned on. In other words, it copies only the files that have been changed since the last backup. This backup turns off the archive bits.
- A **differential backup** is identical to an incremental backups, except that it doesn't turn off the archive bits.
- A **daily backup**, also known as a **daily copy backup**, makes copies of all the files that have been changed that day. It does not change the archive bits.

The motivation for having both the incremental and differential backups may not be clear at first glance—they seem so similar as to be basically the same. Incremental seems the better option at first. If a file is backed up, you would want to turn off the archive bit, right? Well, maybe. But there is one scenario where that might not be too attractive. Most backups do a big weekly normal backup, followed by daily incremental or differential

Tech Tip

Customizing Explorer in Windows XP

Windows Explorer / My Computer in Windows XP by default does not show much about files in any view, even when you select Details from the View menu. The Details view is highly customizable, however, and can reveal a phenomenal amount and variety of information about files. To customize your view, alternate-click (right-click) the column bar (the gray bar that says Name, Size, Type, Date Modified, and so forth) to look at the default choices. You'll see everything from Attributes, Owner, Author, and Title, to file-type specific information such as Genre, Duration, and Bit Rate (for music files). If the 15 default extra view options don't get your motor revving, selecting the More option brings up a menu offering 32 view options! For the purposes of this section, click the Attribute box to display file and folder attributes.

Be sure you know the different types of backups, including which ones change the archive bits and which ones do not.

backups at the end of every business day. Figure 20.2 shows the difference between incremental and differential backups.

Notice that a differential backup is a cumulative backup. Because the archive bits are not set, it keeps backing up all changes since the last normal backup. This means the backup files will get progressively larger throughout the week (assuming a standard weekly normal backup). The incremental backup, by contrast, only backs up files changed since the last backup. Each incremental backup file will be relatively small and also totally different from the previous backup file. Let's assume that the system is wiped out on a Thursday morning. How can you restore the system to a useful state? If you're using an incremental backup, you will first have to restore the last weekly backup you ran on Monday, then the Tuesday backup, and then the Wednesday backup before the system is restored to its Thursday morning state. The longer the time between normal backups, the more incremental backups you must restore. Using the same scenario, but assuming you're doing differential instead of incremental backups, you'll only need the weekly backup, and then the Wednesday backup to restore your system. A differential backup will always require only two backups to restore a system (see Figure 20.3). Suddenly, the differential backup looks better than the incremental! On the other hand, one big benefit of incremental over differential is backup file size. Differential backup files will be massive compared to incremental ones.

Incremental

MON	TUE	WED	THU	FRI
Full Backup	All Tuesday Changes	All Wednesday Changes	All Thursday Changes	All Friday Changes

Differential

MON	TUE	WED	THU	FRI
Full Backup	All Changes Through Tuesday	All Changes Through Wednesday	All Changes Through Thursday	All Changes Through Friday

• **Figure 20.2** Incremental vs. differential

Choosing between incremental backups and differential backups is only one factor in choosing how you back up your data. You must also consider your business, your data, your backup hardware, your operating systems, and other factors to create a backup strategy.

Backup Strategies

One of the issues you must address to successfully answer questions on backups and recoverability is how to choose strategies that meet your needs for doing backups and restores in your actual network environment. Your goal is to be able to back up, and then easily restore all the necessary information from all the necessary machines. This can include both servers and workstations. Decisions about backing up multiple machines can revolve around both hardware and time factors. Another issue to consider when planning your backup strategy is what to do with the all-important tapes. Recognize that you are protecting your network not only against viruses and other computer-related disasters, but also *against* fire, flood, and man-made catastrophes. Let's look at a typical network and see how to create a backup strategy.

Shannen works at a small company that has never had any form of reliable backup. The network consists of a single Windows Server 2003 system that acts as a file server for the entire network. The file server stores an Access database—a single .MDB file—constantly updated during the course of the

Thursday: System Crash!	System Crash!		
Incremental Needs:	Full Backup	All Tuesday Changes	All Wednesday Changes
Differential Needs:	Full Backup	All Changes Through Wednesday	

• **Figure 20.3** Restoring from backups business day by almost every user on the network. Each of the client systems also stores a number of mission-critical Word documents.

Shannen first determines what data needs backing up. She needs to back up the entire server operating system—about 1.3GB for the operating system and applications. The Access database is roughly 45MB and hasn't grown much in the past year. She then goes through the many Word documents stored on each user's system, creates a directory called \Home on the file server with a subfolder for each user, and instructs the users to copy their Word documents to their individual subfolders. Combined, the Word documents take up roughly 100MB.

This is an interesting side effect of backups. Creating a backup strategy often forces network administrators to change the way their network stores data. Shannen will also have to enforce a policy (lots of finger wagging and reminder e-mails) for users to save their work to the server. Perhaps a bit of training might also be in order!

Shannen didn't have to force the users to copy their data to the server just for backing up. Many companies sell superb, powerful backup software that will enable the administrator to back up multiple systems in almost any way he or she might imagine. A single backup job can back up entire systems or selectively back up only certain folders on each system. This type of online backup is popular today. You can even find companies that will back up your systems over the Internet for a monthly charge.

Now that Shannen has a grasp on the data to back up, she turns her attention to her backup hardware. She has a DLT tape drive with 30-GB capacity tapes—far more than her needs of less than 2GB with plenty of room for future expansion. This also tells her that a single tape will hold multiple backups—a handy thing to do to save on tapes! Shannen chooses to use a differential backup method. She'll need two tapes: one for the weekly full backup and one for the daily changes.

Shannen has two more issues to deal with: offsite backups and tape wear and tear. Backup strategies require some method of rotation of tapes to ensure that one backup is offsite in case of a serious disaster, to prolong tape life, and to retire older tapes. This is called a *tape rotation method* or a *tape backup scheme*. Shannen chooses to use a tape rotation method called Grandfather-Father-Son. This tried-and-true method has many variations, but the one Shannen uses works like this. The tape she uses for backup on the last Friday of each month is called the grandfather tape. She stores this tape offsite. The tape she uses every Friday (except the last Friday of the month) is the father tape and is also stored offsite. The tape used for the daily backup (except on Friday) is called the son. Shannen keeps these tapes onsite. If a fire wipes out the office, she'll have a complete server backup that is always less than a week old offsite.

Tech Tip

Full System Backups

Ever wish you could back up your entire system? Sure, Windows has plenty of recovery tools, but they're not always perfect. Tools such as System Restore and the recovery console are useless if your hard drive dies. In some companies, entire systems are backed up using a number of products. These backups are often stored on optical media or on massive server drives on the network. Although full system backups can be complex to use and challenging on your network bandwidth, the ease of system recovery makes this approach popular for those "I need my computer back NOW!" situations.

Try This!

Tape Rotation Methods

Grandfather-Father-Son is only one of many tape rotation methods. Do a web search for tape rotation methods such as Grandfather-Father-Son, Tower of Hanoi, and Incremental. Don't confuse Incremental tape rotation with the incremental backups—they aren't the same thing! What are the benefits and disadvantages of these different methods? Can you come up with situations where one tape rotation method might have advantages over another?

Baselines

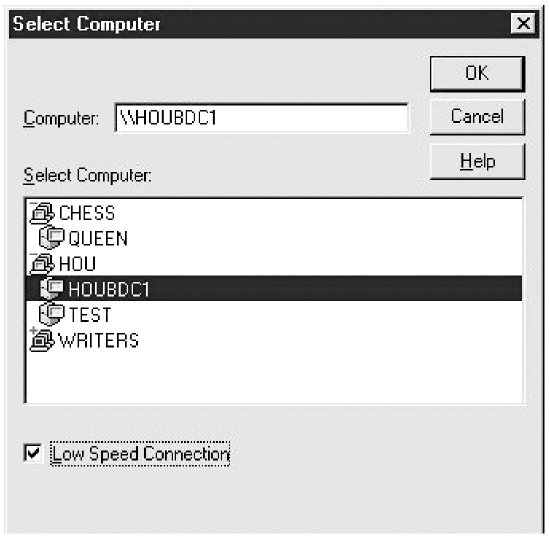
The best way to know when a problem is brewing is to know how things perform when all's well with the system. You need to establish a *baseline*: a log of performance indicators such as CPU usage, network utilization, and other values to give you a picture of your network and servers when they are working correctly. A major change in these values can point to problems on a server or the network as a whole. A common tool used to create a baseline on Windows systems is the Performance Monitor utility that comes with Windows NT/2000/XP/2003. Conveniently, all network operating systems come with similar tools, and you can also create baselines using most network management utilities.

Windows NT/2000/XP/2003—Performance Monitor

Administrators use **Performance Monitor** (also called **PerfMon**) to view the behavior of hardware and other resources on NT/2000/XP/2003 machines, either locally or remotely. PerfMon can monitor both real time and historical data about the performance of your systems. To access the Performance Monitor applet, choose Start | Programs | Administrative Tools | Performance Monitor from any Windows NT machine. On a Windows 2000 system, choose Start | Settings | Control Panel, and then double-click Administrative Tools, and double-click Performance.

Once you access PerfMon, you need to configure it to display data—but first, you must understand the concepts of objects, counters, and views. An **object**, in Performance Monitor terms, is the component of your system you want to monitor, such as the processor or the memory. Each object has different measurable features, called **counters**. Counters, in other words, are the aspects of an object that you want to track. As you decide which object(s) to monitor, you must also select at least one counter for each object. Performance Monitor can organize and display selected counter information using a variety of **views**, each of which provides different types of information. The Log view, for example, enables you to store data about your system for later review. This is the view you use to create a baseline. Although it's the only one I'm going to discuss here, the other views—Chart, Alert, and Report—are useful for troubleshooting problems as they arise.

To access the Log view in Windows NT, either click the Log View button or choose View | Log. To add objects to the Log view, either click Add To (the plus sign) or choose Edit | Add To Log. In the Add To Log dialog box, first select the computer you want to monitor. You can choose either the local machine (the default) or a remote machine. To monitor a remote machine, type the Universal Naming Convention (UNC) name of the computer in question. To monitor a machine named HOUBDC1, for example, you would type **\\HOUBDC1** in the Computer field. You can also use the Select Computer button (at the right end of the Computer field) to view the available machines and select the one you want to monitor, as shown in Figure 20.4.

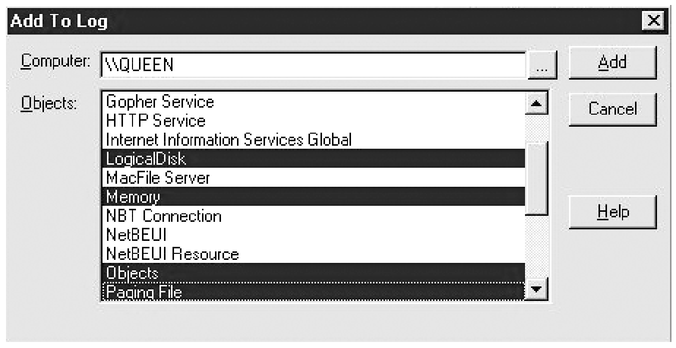


Computer: \\HOUBDC1

☑ Low Speed Connection

• **Figure 20.4** The Select Computer dialog box in Performance Monitor

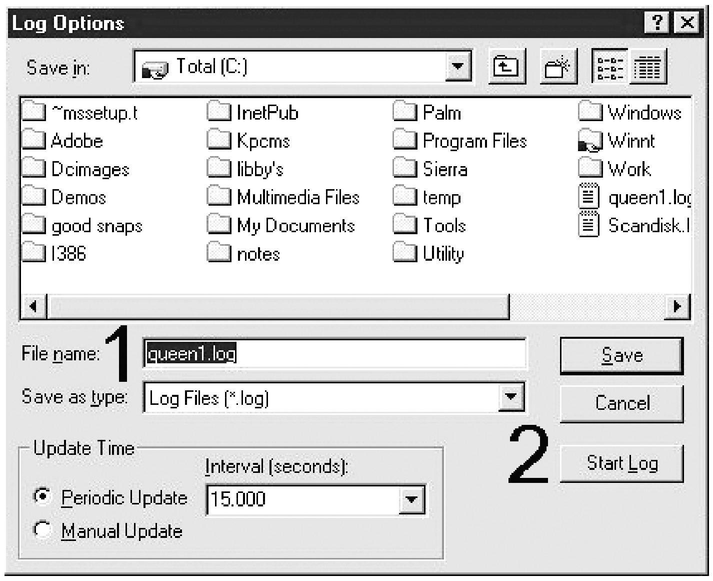
Although it is usually easier to monitor a machine locally, it is often more accurate to monitor a machine remotely. PerfMon itself, as it runs on a machine, uses a certain amount of that system's resources to take the measurements and display the data graphically. You can corrupt your results by monitoring locally, especially when you need to troubleshoot problems with disk performance, memory and paging, or processor use. There are some cases where monitoring locally is preferred or required,


Add To Log
Computer: \\QUEEN
Objects: Gopher Service, HTTP Service, Internet Information Services Global, LogicalDisk, MacFile Server, Memory, NBT Connection, NetBEUI, NetBEUI Resource, Objects, Paging File

• **Figure 20.5** Add To Log in Performance Monitor

however. If you are monitoring network access or networking protocol objects, for example, monitoring locally will affect the readings to a lesser degree than monitoring remotely. Similarly, you must monitor a system locally if you can't access that system over the network. Finally, when you monitor objects created by a specific application, such as Exchange, you should monitor locally, as the objects related to this application are only created locally and will not be available from another system.

Once you have selected a system to monitor, either locally or remotely, you must select the object to monitor. Select one or more objects to monitor from the list in the Object field. Note that the Log view is somewhat different from the other views, in that you only add *objects* to the view, not the specific counters for the objects, as shown in the Add To Log dialog box in Figure 20.5.

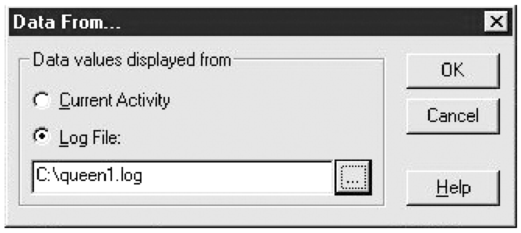

Log Options
Save in: Total (C:)
~mssetup.t, Adobe, Dcimages, Demos, good snaps, I386, InetPub, Kpcms, libby's, Multimedia Files, My Documents, notes, Palm, Program Files, Sierra, temp, Tools, Utility, Windows, Winnt, Work, queen1.log, Scandisk.l
File name: queen1.log
Save as type: Log Files (*.log)
Update Time
Interval (seconds): 15.000
Periodic Update (selected)
Manual Update

• **Figure 20.6** Select a log file (1) and start logging (2) in Performance Monitor.

After you select the objects for the Performance Monitor to track and log, select Options | Log Options to save the data to a log file, and then start logging by clicking the Start Log button, as shown in Figure 20.6. This dialog box also gives you the opportunity to select the update method and time.

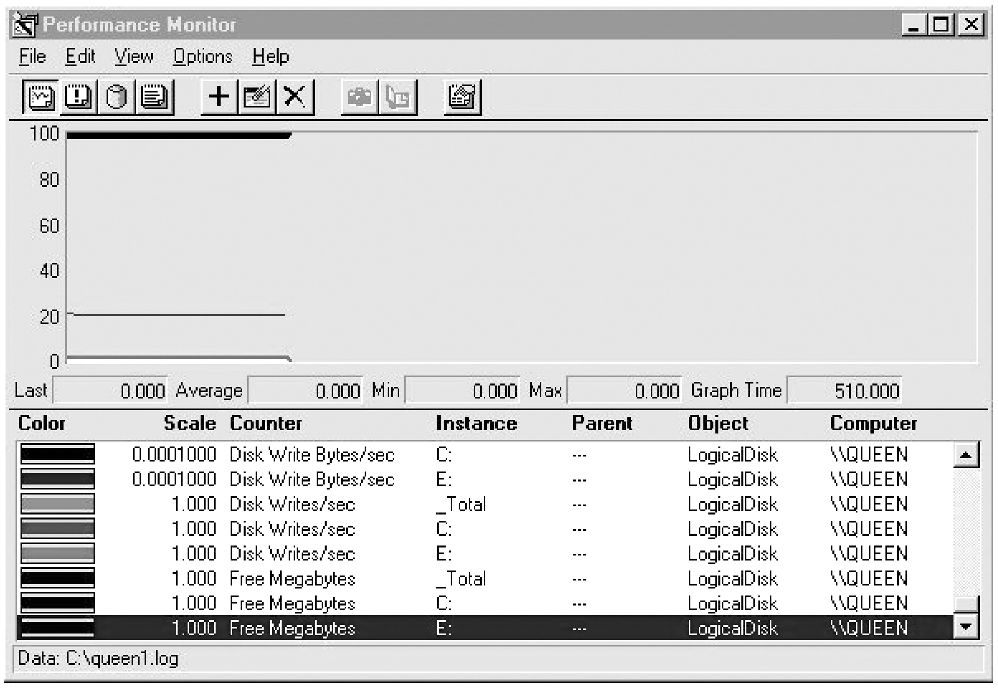
After you have configured the log to save to a particular file, you can see the log file name, status of the logging process, log interval, and file size of the log in the Performance Monitor dialog box. To stop collecting data in a log, open the Log Options dialog box again and click Stop Log. You can then choose to create a new Log file and begin logging again, if necessary. You can also view data from one of these saved log files by selecting Options | Data From. In the Data From... dialog box, shown in Figure 20.7, you can choose to continue obtaining data from the current activity, or obtain data from a particular log file.

When you choose to obtain data from a saved log, you go back to that frozen moment in time and add counters to the other views for the objects you chose to save in the log. In our Log options, for example, we chose to store data for the Logical Disk object. After we've loaded that particular log file, we can change to the Chart view, add counters there for the Logical Disk object, and view a static chart for that moment in time (see Figure 20.8). You may want to select a wide variety of objects to save while in Log view, so that when you open the log to display in any of the other views (Chart, Alert, and Report), you can add a wide range of counters.


Data From...
Data values displayed from
Current Activity
Log File: (selected)
C:\queen1.log

• **Figure 20.7** The Data From... dialog box in Performance Monitor

The Performance Monitor utility described here is specific to Windows NT systems, but you should create baselines for whatever types of systems you have, and they should cover all aspects of your network. Be certain to create multiple baselines, to show the



Last 0.000 Average 0.000 Min 0.000 Max 0.000 Graph Time 510.000

Color	Scale	Counter	Instance	Parent	Object	Computer
■	0.0001000	Disk Write Bytes/sec	C:	---	LogicalDisk	\\QUEEN
■	0.0001000	Disk Write Bytes/sec	E:	---	LogicalDisk	\\QUEEN
■	1.000	Disk Writes/sec	_Total	---	LogicalDisk	\\QUEEN
■	1.000	Disk Writes/sec	C:	---	LogicalDisk	\\QUEEN
■	1.000	Disk Writes/sec	E:	---	LogicalDisk	\\QUEEN
■	1.000	Free Megabytes	_Total	---	LogicalDisk	\\QUEEN
■	1.000	Free Megabytes	C:	---	LogicalDisk	\\QUEEN
■	1.000	Free Megabytes	E:	---	LogicalDisk	\\QUEEN

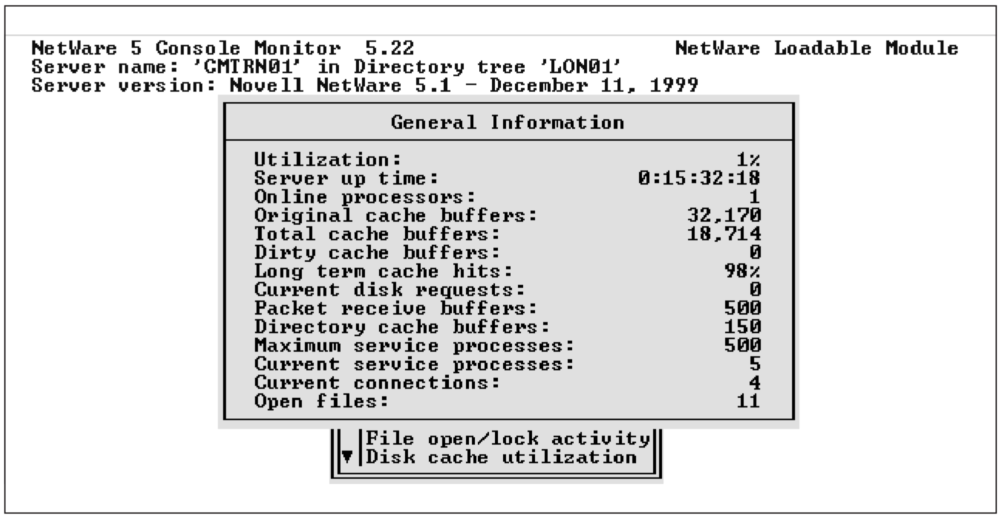
Data: C:\queen1.log

• **Figure 20.8** Viewing a static chart in Performance Monitor

systems both at rest and in use, using the Performance Monitor as well as other systems management or network sniffer tools.

NetWare—Console Monitor

On a NetWare server, most of the critical information you might need to see and document to establish your baseline can be obtained by loading the **Console Monitor** application (see Figure 20.9) on the server itself. You can view the program remotely on a client PC, but it only runs on the server. Novell NetWare calls a program that runs on the server in this way a



```
NetWare 5 Console Monitor  5.22                    NetWare Loadable Module
Server name: 'CMTRN01' in Directory tree 'LON01'
Server version: Novell NetWare 5.1 - December 11, 1999

                          General Information

                  Utilization:                              1%
                  Server up time:                  0:15:32:18
                  Online processors:                         1
                  Original cache buffers:               32,170
                  Total cache buffers:                  18,714
                  Dirty cache buffers:                       0
                  Long term cache hits:                    98%
                  Current disk requests:                     0
                  Packet receive buffers:                  500
                  Directory cache buffers:                 150
                  Maximum service processes:               500
                  Current service processes:                 5
                  Current connections:                       4
                  Open files:                               11

                          File open/lock activity
                          Disk cache utilization
```

• **Figure 20.9** The NetWare 5 Console Monitor general information screen

NetWare Loadable Module (NLM). You issue the command **LOAD MONITOR** at the server's console prompt to start the program.

The Console Monitor NLM can display a wide range of information, from memory use to individual statistics about the network interface cards (NICs) installed in the server. Many system managers leave Console Monitor running all the time so they can keep an eye on things. It can also be used to kick users off the server and see which files they're accessing!

Using Baselines

Using baselines requires a serious commitment. Not only do you need to create them in the first place, you need to monitor them continually to know when changes take place. Be careful not to be fooled by a bad baseline—make a number of baselines at first to get an idea as to when your network is most quiet and when it's most busy. Many administrators like to make dual baselines: one for when the network/server is at rest and another when the network/server is at its busiest. This is a matter of personal choice—try both the single baseline and the dual baseline to see which you prefer.

Troubleshooting Model

No matter how complex and fancy, any **troubleshooting model** can be broken down into simple steps. Having a sequence of steps to follow makes the entire **troubleshooting process** simpler and easier, because you have a clear set of goals to achieve in a specific sequence. The most important steps are the first three—they help you narrow down the cause of the problem to a specific item. The reason this matters so much is that figuring out what's wrong will also probably tell you how to fix the problem, and how to prevent it from happening in the future.

The basics of any troubleshooting model should include the following steps:

1. Establish the symptoms.
2. Isolate the cause of the problem (identify the scope of the problem).
3. Establish what has changed that might have caused the problem.
4. Identify the most probable cause.
5. Implement a solution.
6. Test the solution.
7. Recognize the potential effects of the solution.
8. Document the solution.

Establish the Symptoms

If you are working directly on the affected system and not relying on somebody on the other end of a telephone to guide you, you will establish the symptoms through your observation of what is (or isn't) happening. If you're troubleshooting over the telephone (always a joy, in my experience), you will need to ask questions based on what the user is telling you. These questions can be *close-ended,* which is to say there can only be a yes or no type answer, such as, "Can you see a light on the front of the monitor?" You can also ask *open-ended* questions, such as, "Tell me what you see on the screen." The type of question you use at any given moment will depend on what information you need, and on the knowledge level of the user. If, for example, the user seems to be technically oriented, you will probably be able to ask more close-ended questions, because they will know what you are talking about. If, on the other hand, the user seems to be confused about what's happening, open-ended questions will allow him to explain what is going on in his own words.

Isolate the Cause of the Problem

One of the first steps in trying to determine the cause of a problem is to understand the extent of the problem—is it specific to one user or is it network-wide? Sometimes this entails trying the task yourself, both from the user's machine and from your own or another machine.

For example, if a user is experiencing problems logging in to the network, you might need to go to that user's machine and try to use their user name to log in. This will tell you whether the problem is a user error of some kind, as well as enable you to see the symptoms of the problem yourself. Next, you probably want to try logging in with your own user name from that machine, or have the user try to log in from another machine. In some cases, you can ask other users in the area if they are experiencing the same problem, to see if the problem is affecting more than one user. Depending on the size of your network, you should find out if the problem is occurring in only one part of your company, or across the entire network.

What does all of this tell you? Essentially, it tells you how big the problem is. If nobody in an entire remote office can log in, you may be able to assume that the problem is the network link or router connecting that office to the server. If nobody in any office can log in, you may be able to assume that the server is down or not accepting logins. If only that one user in that one location can't log in, it may be a problem with that user, that machine, or that user's account.

Eliminating variables is one of the first tools in your arsenal of diagnostic techniques.

Establish What Has Changed That Might Have Caused the Problem

After determining the extent of a problem, the next step is to eliminate all the extra variables, that is, all the incorrect possible causes of the problem. If you have determined that the problem is specific to that user on that machine, you have already learned a great deal. First, you have learned that this isn't a user account problem, because that user was able to log in from another machine. You have also determined that it isn't user error, because you've tried it yourself. By having other users at other machines successfully try the task, you eliminated the possibility that the server is down.

Ask Isolating Questions

The goal of this step is to isolate the problem to a specific item (hardware, software, user error, and so on), or to identify what has changed that might have caused the problem. You may not have to ask many questions before the problem is isolated, but it can sometimes take quite a bit of time and involve further work behind the scenes. Isolating questions are designed to home in on the likely cause of the problem. Here are some examples:

- "Tell me exactly what you were doing when the problem occurred."
- "Has anything been changed on the system recently?"
- "Has the system been moved recently?"

Notice the way I've tactfully avoided the word *you,* as in "Have *you* changed anything on the system recently?" This is a deliberate tactic to avoid any implied blame on the part of the user. Being nice never hurts and it makes the whole troubleshooting process more friendly.

You should be asking some isolating questions *internally* of yourself, such as, "Was that machine involved in the software push last night?" or "Didn't a tech visit that machine this morning?" Note that you will only be able to answer these questions if *your* documentation is up to scratch. Sometimes, isolating a problem may require you to check system and hardware logs (such as those stored by some routers and other network devices), so make sure you know how to do this.

Identify the Most Probable Cause

This step comes down to experience—or good use of the support tools at your disposal, such as your knowledge base. You need to select the most *probable* cause from all the *possible* causes, so the solution you choose fixes the problem the first time. This may not always happen, but whenever possible, you want to avoid spending a whole day stabbing in the dark while the problem snores softly to itself in some cozy, neglected corner of your network.

Implement a Solution

Once you think you have isolated the cause of the problem, you should decide what you think is the best way to fix it, and then try your solution, whether that's giving advice over the phone to a user, installing a replacement part, or adding a software patch. All the way through this step, try only one likely solution at a time. There's no point in installing several patches at once, because then you can't tell which one fixed the problem. Similarly, there's no point in replacing several items of hardware (such as a hard disk and its controller cable) at the same time because then you can't tell which part (or parts) was faulty. As you try each possibility, always *document* what you do and what results you get. This isn't just for a future problem, either—during a lengthy troubleshooting process, it's easy to forget exactly what you tried two hours before, or which thing you tried produced a particular result. Although it may take longer to be methodical, it will save time the next time—and it may enable you to pinpoint what needs to be done to stop the problem from recurring at all, thereby reducing future call volume to your support team—and as any support person will tell you, that's definitely worth the effort!

Remember these problem analysis steps:

- Establish the symptoms
- Isolate the cause
- Establish what has changed
- Ask isolating questions
- Identify the most probable cause
- Implement a solution
- Recognize potential effects of the solution
- Document the solution

Test the Solution

This is the part everybody hates. Once you think you've fixed a problem, you should try to make it happen again. If you can't, great! But sometimes you will be able to re-create the problem, and then you know you haven't finished the job at hand. Many techs want to slide away quietly as soon as everything seems to be fine, but trust me on this, it won't impress your customer when their problem flares up again 30 seconds after you've left the building—not to mention that you get the joy of another two-hour car trip the next day to fix the same problem, for an even more unhappy client! In the scenario where you are providing support to someone else rather than working directly on the problem, you should make *them* try to re-create the problem. This will confirm whether they understand what you have been telling them, and will educate them at the same time, lessening the chance they'll call you back later and ask, "Can we just go through that one more time?"

Try This!

Mnemonic for Troubleshooting

How will you remember these eight troubleshooting steps? Create a mnemonic for the steps that you can use to cement them into your brain. Pick a theme, such as food or colors, and make the mnemonic—the memory tool—work. Don't be afraid to be silly or funny, just go for something memorable!

Recognize the Potential Effects of the Solution

Okay, now that *you* have changed something on the system in the process of solving one problem, you must think about the wider repercussions of what you have done. If you've replaced a faulty NIC in a server, for instance, will the fact that the MAC address has changed (remember, it's built into the NIC) affect anything else, such as the logon security controls, or your network management and inventory software? If you've installed a patch on a client PC, will this change the default protocol or any other default settings that may affect other functionality? If you've changed a user's security settings, will this affect their ability to access other network resources? This is part of testing your solution to make sure it works properly, but it also makes you think about the impact of your work on the system as a whole.

Document the Solution

It is *vital* that you document the problem, symptoms, and solutions of all support calls, for two reasons. First, you're creating a support database that will be a knowledge base for future reference, enabling everyone on the support team to identify new problems as they arise, and know how to deal with them quickly, without having to duplicate someone else's research efforts. Second, documentation enables you to track problem trends and anticipate future workloads, or even to identify a particular brand or model of an item, such as a printer or a NIC, that seems to be less reliable or that creates more work for you than others. Don't skip this step—it *really* is essential!

Troubleshooting as Art—Mike's Four-Layer Model

Troubleshooting is not something that can be definitively described in a nice neat list of ten easy steps. It is more of an art—an ability to become "one with the network" and intuit where the problems are hiding. The best troubleshooters are those who have a huge amount of knowledge about every element of the network—hardware, software, connections, and so on. These people can then synthesize all that knowledge into good guesses about where to start looking for problems. All of the previous steps give you a theoretical concept of where to look and how to proceed when troubleshooting your own network. The theory, however, is easier to implement in real life if you know where (and where not) to look within the network to find the problem.

The troubleshooting model does a great job of describing the steps necessary to get a network problem fixed, but it doesn't tell you where to look in the first place. So how do you figure out where to look for problems? Some might think to use the OSI seven-layer model, but while the OSI seven-layer model provides a superb tool for those who create network hardware and software, it doesn't do a lot for the folks who need to work on a network every day. Instead, I use a model that I modestly call Mike's Four-Layer Model. The Four-Layer Model concentrates on the main components of the hardware and software you need to deal with to make a networking system work, while ignoring the parts of the hardware and software over which you have no control. The core idea behind the four layers of my model is that you can access, change, remove, install, troubleshoot, and fix every layer—unlike most of the layers of the OSI seven-layer model.

The Four-Layer Model breaks all networks into—surprise!—four major areas: Hardware, Protocols, Network, and Shared Resources. No matter what specific networking hardware or software you use, every one of these four layers exists—although some network operating systems like to hide them! If you understand Mike's Four-Layer Model, you should have no problem fixing just about any network, any time. By using the Four-Layer Model when you have a problem with a network, you'll never find yourself saying, "I have no idea why this is happening to my network! What do I do?" Instead, you'll find yourself saying, "I know why this problem exists. I just need to figure out how to fix it!" Trust me, figuring out how to fix a known problem is a lot easier than trying to troubleshoot a mystery problem on your network.

The layers of Mike's Four-Layer Model are

1. Hardware
2. Protocols
3. Network
4. Shared Resources

Mike's Four-Layer Model also *ignores the function of the layers* and concentrates instead on *what you need to do to make the layers function* (that is, what you do to verify that a particular layer works correctly or needs diagnosing). For example, if you've installed the hardware correctly, then the hardware layer works. To determine this, you run a certain type of test (like pinging another system), and if you're successful, you in essence check off the hardware layer as good and move on to the next one.

Some of my critics suggest that if diagnostic success is the goal, then my Four-Layer Model is by definition not a true model; they argue that it's nothing more than a diagnostic procedure. These people do have a point—if you get technical about it, my Four-Layer Model *is* nothing more than a diagnostic procedure. Here's what I have to say to those critics: "Well, it's the best darn totally-universal, easy-to-understand, 100-percent-accurate, *practical* diagnostic procedure you'll ever meet! Six weeks after the Network+ exam, you'll probably have forgotten most of the seven OSI layers, but you'll never forget my model!" Why? Because you'll use it every day!

I once had a reader e-mail me about the Four-Layer Model. She said I should change its name to Mike's Four-Doors Model. Her suggestion for the name change came from the fact that the four layers of my model seemed to her like doorways to the innards of the network, rather than a true definition of all network functions. In a way, she was right—my model does not define all the functions of a network, like the OSI seven-layer model does. I ignore or blend most of the OSI layers into my model because I *want to deal with things I can touch in the network*. So, I guess the doorway analogy is a decent way to think about my model, but I'm too lazy to change all my work so I'm sticking to the name Four-Layer Model. Let's look at each of its layers.

Hardware

Hardware is probably the most self-explanatory of the four categories. This layer covers the many different ways data can be moved from one PC to another, from copper to fiber to wireless signals. It includes the Physical and Data Link components of the OSI model, but concentrates on the parts that make this all work, like hubs, cables, and connectors. It also includes items that the OSI model does not directly address, like NICs, device drivers for NICs, and how you install and test them.

Mike's Four-Layer Model is in fact nothing more than this one tech's opinion of how best to fix a network. It is neither an industry standard nor that well-known—yet. It is not part of the Network+ test in any way, but it will help you on the test by providing a framework you can use to analyze the many scenario questions on the exam and quickly generate correct answers.

Protocols

I should call this section "Network Protocols," but I don't want you confusing it with the next one. Protocols are the languages of networks. As you've learned, these languages have interesting names such as NetBIOS/NetBEUI, IPX/SPX, and the ever-popular TCP/IP. A *protocol* is a highly standardized language that handles most of the "invisible" functions on a network, like determining which computer is SERVER1, or disassembling and reassembling data passed over the network. Here we concern ourselves with installing and configuring the protocol so that a system can communicate with other systems on the network. We're not too interested in how this all works in great detail—the point here is that if you install the right protocol and configure it properly, it will work.

Network

Once you install the hardware and the protocol, you need to make two critical determinations. First, you need to decide which systems will share resources and which systems will only access those shared resources. In other words, you must determine which systems will act as servers and which will act as clients, and then configure them accordingly. Second, you need to name the systems, so they can see each other, usually something like *Server1* or *Mike's PC.* In this layer, the concepts of client/server, peer-to-peer, and domain-based networks come into play. Different network operating systems use different methods to name their systems and to determine which systems share and which do not. This layer requires that you appreciate the differences among programs like Windows, NetWare, and Linux.

Shared Resources

The entire reason we network is to share resources, so this final layer encompasses all the steps required to enable a system to share a resource, and to allow and enable other systems access to that shared resource. This layer includes a number of steps, and unlike the earlier layers, these steps often need to be done fairly frequently, because by their nature, shared resources and the users who access them change pretty much continuously. This layer includes most of the day-to-day administration of networks and tends to be the place we spend the majority of our time.

Using the Four-Layer Model

The secret to using the Four-Layer Model to diagnose network problems is to use it to structure your analysis when problems occur, and to proceed through the layers as you diagnose the problem. Let me show you how I use it by giving you a scenario.

I recently replaced an ancient router that had an ISDN Internet connection, with a new SOHO gateway DSL router, in a Windows 2000 network. The Windows network had DNS, WINS, and DHCP servers for all the clients on the network. I configured the new router to use the same IP addresses as the old router. When I went to a system to test the setup, I popped right up on the Internet, no problem. I then fired up IPCONFIG and released and renewed the IP address to make sure the new router worked. I got a new IP address and everything looked okay. Then I decided to browse the network—and I couldn't see any other system! Not good. Time to use the Four-Layer Model.

First, the Physical layer—was I disconnected from the network, or was a piece of hardware broken? Not likely, given the fact that I could still get on the Internet. But I used PING to test the router—no problem. Then I pinged the server and didn't get an answer. Hmm, was there a cable problem between me and the server? I checked and saw good link lights all around. Plus, everyone else was still hitting the server just fine. Time to assume the Physical layer was okay.

On to the second layer, Protocol. I'm running TCP/IP, and it's unlikely I could get to the Internet if I had a problem there. But I still checked the system with IPCONFIG. Wait a minute! A quick look reveals that my IP address is 192.45.15.10, but the network ID I want is 192.168.4.0! This is weird—where did this strange IP address come from? Well, I set up the router for DHCP—maybe there's a problem with the DHCP server. The network admin checks and tells me it's up and running. I want to verify this so I go to another system and do a release/renew there. Lo and behold, suddenly IT has an IP address of 192.45.15.11! What's going on here? I ask myself the million-dollar question: "What has changed on this network that might have caused this?" The answer was easy: the only recent change to the network was the new router. A quick examination reveals the problem: The router was set up to do DHCP. I quickly shut off the router's DHCP, and then do another release/renew on the two systems with the bad IP addresses, and voilà, problem solved, and I'm on to the next job!

By using the Four-Layer Model, I could quickly remove physical issues from the possible problems and move on to examine protocol issues, where I found the problem. Try my Four-Layer Model for yourself—it works!

Tech Tip

WINIPCFG

If you have Windows 9x/Me, you can run the graphical utility WINIPCFG in place of IPCONFIG. It accomplishes the same goals, but like all good GUI tools, it gives you buttons to click instead of command strings to type. To run the application, go to Start | Run and type **WINIPCFG**; *click the OK button and you're off!*

■ Troubleshooting Scenarios

I want to end this chapter and the book with some good troubleshooting scenarios. Take some time and think about these situations and how you would handle them. What questions would you ask? What tests would you do first? The Network+ exam absolutely *loves* to ask scenario questions. The knowledge from the previous chapters combined with the methods you've learned in this chapter should enable you to fix any network!

"I Can't Log In!"

One of the most complex troubleshooting issues is that one set of symptoms, in this case a user's inability to log in, can have many causes. Suppose Woody has called complaining that he cannot log in to the company's intranet. Tina

Tech first tries accessing the intranet site from her workstation, and finds she has no problem. Tina might also want to have other users try to log in, or confirm that other users are not having the same problem. Next, Tina should have Woody try to log in from another machine. This will help Tina determine whether the problem lies with Woody's user account's capability to log in, or with Woody's Windows 98 workstation or connectivity.

If Woody is unable to log in from another machine, Tina should probably check to be sure Woody is using the correct login ID, password, and procedure when he logs in. On the other hand, if Woody is able to log in from another user's workstation, Tina should probably focus on determining whether Woody's workstation is working properly and connecting to the network. One step she could try here is pinging Woody's workstation. If Tina is able to ping Woody's machine successfully, she knows that the machine is up, the TCP/IP protocol is configured correctly, and the system is connected to the network. Tina might then check the configuration of the network client on Woody's workstation. If Tina is not able to ping the workstation, however, she might need to test the cables and NIC using cable testers or loopback devices, and verify that TCP/IP was correctly configured using WINIPCFG.

"I Can't Get to This Web Site!"

Reaching external web sites requires that a variety of components be configured correctly. Some of these components are within your company's internal control; many of them are not. When Fatima calls and tells Tina Tech that she cannot reach www.comptia.org, Tina's first step is to try to reach that site herself. In this case, Tina was also unable to get a response from the comptia.org site. One of her next steps is to ping the site, first by name, and then by IP address. In this case, she gets no response by name, but she does get a normal response when she pings the site by IP address. This immediately indicates to her that the problem is name resolution, in this case: DNS.

On the other hand, had Tina been unable to ping successfully using either the IP address or host name, she should consider two possibilities. First, if her company uses a firewall or proxy server to reach the Internet, she should ping that machine. This machine usually has the same IP address as the default gateway TCP/IP setting. If Tina can successfully ping her default gateway, she can be almost certain that the problem is not something she or her company has any control over. To verify this, Tina should attempt to reach some other external sites, both by pinging and using a web browser. If she can reach other sites successfully, the problem is most likely with the comptia.org site or the gateway.

"Our Web Server Is Sluggish!"

Slow response from a server can be related to a variety of things. Usually, however, the problem can be traced to a connection to the server, or to the server itself. When Wanda calls in from working at home and tells Tina Tech that she is getting a slow response from the company's web site, Tina Tech leaps into action. Tina tries to reach the offending server and is immediately connected; this indicates a connectivity problem for that user. She asks Wanda to execute a **TRACERT** command from her workstation to the slow server. This reveals to Tina that the slowdown stems from one of the intermediate steps through which Wanda's system connects to the server. Because of this, the problem is out of Tina's hands, unless she can offer a direct dial-up option for Wanda.

If Tina finds she cannot reach the offending server quickly when she tries from her workstation, then the problem may lie with the server itself. Tina checks the Change Log for the web server, to see if anyone has changed anything recently. She discovers a new anti-virus component was recently added, so she checks the vendor's web site to make sure there are no known problems or patches for that piece of software. She also uses the Performance Monitor to compare the server's current responses to the baseline that she previously recorded. This shows her that the bottleneck is related to excessive paging, indicating that the server may need more physical memory, or RAM.

"I Can't See Anything in Network Neighborhood!"

When a user is completely cut off from the network, the problem is usually limited to that user's workstation or network connection. When Tina gets a call from Johnny saying his Windows 98 machine is on, but that he can't log in and can't see any other machines on the company's TCP/IP network, Tina goes to Johnny's office to run some tests. The first test Tina runs is to ping an external machine. She doesn't expect it to work, but tests just to be certain. Next, she tries to ping Johnny's machine using either **ping localhost** or **ping 127.0.0.1** (remember the loopback address?). When this ping doesn't work, Tina guesses that the problem is in the TCP/IP configuration. To view the machine's TCP/IP configuration, Tina uses WINIPCFG, and notices the IP address is blank. After checking her network documentation to verify what IP address Johnny's machine should have, she adds the IP address and he is able to connect to the network.

If Tina's **ping 127.0.0.1** had worked, she would have had to assume the TCP/IP and networking configuration of Johnny's machine was correct. She should then check the hardware, using a network card utility to verify that the NIC itself is working correctly, and a cable tester to verify that the cable from Johnny's workstation is operating properly. In this case, the cable tester shows that the cable is bad, so she replaces the cable between Johnny's workstation and the patch panel, and he is able to connect.

Troubleshooting Is Fun!

The art of network troubleshooting can be a fun, frolicsome, and frequently frustrating feature of your network career. By applying a good troubleshooting methodology and constantly increasing your knowledge of networks, you too can develop into a great troubleshooting artist. This takes time, naturally, but stick with it. Begin the training. Use the Force. Learn new stuff, document problems and fixes, talk to other network techs about similar problems. Every bit of knowledge and experience you gain will make things that much easier for you when crunch time comes and a network disaster occurs—and as any experienced network tech can tell you, it will, even in the most robust network.

Chapter 20 Review

Chapter Summary

After reading this chapter and completing the exercises, you should understand the following about networking.

Describe troubleshooting tools

- Network troubleshooting tools can be separated into three categories: software tools, hardware tools, and "touchy" tools. The touchy tools include knowing what questions to ask end users, in a thoughtful way, regarding their computer problems. Hardware tools include cable testers, protocol analyzers, hardware loopback devices, and toners. Software tools include protocol analyzers, network analyzers, system logs, performance monitors, and system management software suites. The best "toolbox" to carry with you at all times is your brain.

Explain the troubleshooting process

- The troubleshooting process is a dynamic fluid process that requires quick decisions to get the network going again. Be conservative in your approach—protect your network's data with two basics—keep regular backups, and create baselines for later comparisons. Be aware that each file on your systems has an associated archive bit, which indicates that a file has changed. Know the different type of backups, as well as which backups clear and which backups set the archive bit on.
- A full (or normal) backup copies everything and clears the archive bits. A copy backup is a true copy of an earlier backup, and does not affect the archive bits. An incremental backup (basically, a copy of all files that have changed since the most recent backup) copies only those files that have the archive bit set, and then clears the archive bit on just those files. A differential backup (capturing those files that have changed since the last full backup) is like an incremental backup, except it doesn't affect the archive bit. Finally, a daily (or *daily copy*) backup makes a copy of all the files that have changed that day, and does not change the archive bit. Frequency, storage, and type of backups performed are major decisions that involve careful planning and analysis—establishing just how much data the organization is willing to risk versus how much time and storage it's willing to devote to backups.
- Baselines create a benchmark or standard measurement of when everything is working correctly on your network. Consider this to be a snapshot of your systems. In Windows NT/2000, the Performance Monitor (PerfMon) watches Windows NT/2000/XP/2003 machines, either locally or remotely. Know the concepts of objects (like CPU, memory, or disk drives), counters (which do the actual tracking), and views (which allow real-time or historical arrangements of collected data). Novell NetWare servers have a loadable Console Monitor application. Although Console Monitor can be viewed on a remote system, it only runs on the NetWare server. These are called NetWare Loadable Modules (NLMs), and this particular one is started with the LOAD MONITOR command at the server console prompt.
- Any troubleshooting model should include the following simple steps: start out by establishing the symptoms, isolating the cause and scope of the problem (this includes asking end users the right kinds of isolating questions), and determining what has changed that might have caused the problem. Then you can identify the most probable cause, implement a solution, test the solution, and recognize the potential effects of the solution. Wrap things up with proper documentation during your troubleshooting—this will help you keep track of all that you've tried, and also come in handy in the future. Those who do not document now have no right to complain about lack of documentation in the past!
- "Mike's Four-Layer Model" approaches troubleshooting as an art form. The four layers suggested are hardware, protocols, network, and shared resources. Mike's hardware layer involves, obviously enough, making sure the right hardware is in place, powered on, and connected. Mike's protocol layer involves making sure configuration settings are correct. It's always best to fully test out any changes in equipment to avoid being called back to the scene to finish things off properly.

Mike's network layer involves making sure network devices, especially servers, are functioning normally. Finally, Mike's shared resources layer involves checking for adequate permissions/rights and shares on resources to provide appropriate access to network clients.

Analyze troubleshooting scenarios

- Some of the biggest and most complex troubleshooting scenarios involve having users simply log on. Trying to log on from other nearby machines will help to isolate and define the problem. Having yourself attempt to log on and try the PING command are further steps to help solve this situation. Another problem that pops up occasionally is when a user cannot get to a certain web site. Have the user try to reach another web site like www.CompTIA.org and try to ping the site as well. You may then be able to determine whether DNS is functioning properly or not. Quite simply it could be their server that is temporarily down, too.
- Another common complaint from end users is that server response times are slow. Comparisons against preestablished baselines, mentioned earlier, will yield clues as to actual versus perceived slowdowns. The TRACERT command could also reveal a slow hop along the data packet's path.
- A final problem mentioned by many end users that involves solid troubleshooting is when Network Neighborhood/My Network Places does not show other machines—a connectivity issue. Use the PING command to ping the loopback address (127.0.0.1 or LocalHost), ping the system's own IP address, ping the default gateway (normally a router), and then ping a remote machine. The PING command helps you determine where the connection drops off, enabling you to further track down the item at fault.
- Remember as you practice the Art of Network Troubleshooting to keep it fun, learn all you can, and have an "up" day! Good luck on your Network+ exam!

Key Terms

Archive bit *(534)*
Backups *(533)*
Baselines *(533)*
Console monitor *(539)*
Copy backup *(534)*
Counter *(537)*
Daily backup (daily copy backup) *(534)*
Differential backup *(534)*
Hardware tools *(530)*
Incremental backup *(534)*
Mike's Four-Layer Model *(533)*
NetWare Loadable Module (NLM) *(540)*
Network analyzer *(531)*
Network Monitor (NetMon) *(531)*
Normal backup *(534)*
Object *(537)*
Performance Monitor (PerfMon) *(537)*
Protocol analyzer *(531)*
Software tools *(531)*
System logs *(531)*
Touchy tools *(529)*
Troubleshooting model *(540)*
Troubleshooting process *(540)*
Troubleshooting scenarios *(530)*
Troubleshooting tools *(531)*
View *(537)*

Key Term Quiz

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

1. The term ________________ is used to describe knowing what questions to ask end users.
2. Having access to ________________ is important, although you do not necessarily need to carry around a full toolbox.
3. The best Windows NT/2000 software tool to use when trying to establish baselines is ________________.
4. Another name for a full backup is a ________________.
5. A(n) ______________ backup copies only the files that have changed since the most recent backup and clears the archive bit on those files.

6. A(n) ________________backup copies only the files that have changed since the most recent backup and does not clear the archive bits.
7. The ________________ indicates that a file has been changed and no backup currently exists for that version of the file.
8. The ________________ is a NetWare server's equivalent of Windows NT/2000's Performance Monitor.
9. The LOAD MONITOR command will load a ______________ onto a Novell NetWare server.
10. Any ________________ will include initial steps to define and isolate the problem.

Multiple-Choice Quiz

1. What should be done to establish normal operating conditions on your network?
   - **A.** Backups
   - **B.** Baselines
   - **C.** Remote access
   - **D.** Troubleshooting
2. Which of the following should you do first when troubleshooting a network problem?
   - **A.** Isolate the problem
   - **B.** Test the solution
   - **C.** Implement a solution
   - **D.** Document the solution
3. Which of the following would be an appropriate place to keep your hardware toolkit? (Select all that apply.)
   - **A.** In the server room
   - **B.** In the trunk of your car
   - **C.** At home
   - **D.** None—you can get by without one
4. What should you keep in your hardware toolkit? (Select all that apply.)
   - **A.** Cable tester
   - **B.** Crimpers
   - **C.** Spare cable
   - **D.** Games
5. Where could you find a list of problems that occurred on a system?
   - **A.** DNS Server
   - **B.** Event Viewer
   - **C.** User Manager
   - **D.** User Manager for Domains
6. Julie has spent half a day troubleshooting a network problem and has isolated it to (most likely) configuration or hardware issues on a single PC. The rest of the network works fine, but Julie now has a frustrated computer user who needs her PC networked to get work done. Which of the following offers Julie the best chances of getting good information from the frustrated user? (Select two.)
   - **A.** When did you first notice your problem?
   - **B.** When was it working last?
   - **C.** Have there been any changes made to the hardware or configuration recently?
   - **D.** What have you done, you great oaf?
7. Which of the following items should be done last when troubleshooting a network problem?
   - **A.** Isolate the problem
   - **B.** Test the solution
   - **C.** Implement a solution
   - **D.** Document the solution
8. Where would DNS problems likely occur? (Select all that apply.)
   - **A.** DNS server
   - **B.** Hosts file
   - **C.** Lmhosts file
   - **D.** WINS server
9. What is required on a UNIX/Linux server when logging on as root?
   - **A.** Password
   - **B.** Placename
   - **C.** Rootname
   - **D.** User name

10. What must you have to log onto a Windows 2000 Server system as Administrator? (Select all that apply.)

A. Password

B. Place name

C. Root name

D. User name

11. You are called to assist a user who cannot access any shared files on the network. You open the NIC's properties to find that no protocols are installed. What should you do?

A. Format the hard drive

B. Reinstall all programs

C. Reinstall the operating system

D. Reinstall the protocols

12. Where should backup tapes be kept?

A. In the server room

B. On a shelf in the storeroom

C. In a locked cabinet away from the server room

D. In the boss's car trunk

13. Which of the following are levels of "Mike's Four-Layer Model" to troubleshooting? (Select all that apply.)

A. Hardware

B. Protocol

C. Network

D. Shared resources

14. What item can help isolate a bad NIC that is sending out too much traffic?

A. Bus

B. Hub

C. Protocol analyzer

D. Router

15. Being able to ping a remote system by IP address, but not by FQDN, suggests what type of error?

A. DNS

B. Routing

C. NetBEUI

D. Routing

Essay Quiz

1. Some students in class are discussing when tape backups should be performed. One student says daily, during nonpeak hours, while another student suggests weekly. A third student says both students are correct. The trio suddenly looks at you for your definitive answer. Write down what you would say.

2. You are the new network technician at your company working the Monday morning shift for the first time. An end user calls in saying server response times are slow. Write down what you should say to the end user, as well as what you would do.

3. The IT group at your company has been told to do nightly backups as quickly as possible. Backup tapes are the backup method of choice. Jot down the thoughts you would contribute to the discussion.

4. After a recent file restore fiasco at the nonprofit organization you work at, management decides to eliminate the concept of incremental backups. You are asked to choose another backup method that would not involve restoring so many tapes. Write down which method you would use and give reasons as well.

5. You are on your way back from a call to unlock an end user's account by resetting their password. On entering the server room you notice it feels a little warm. Then you notice that a colleague has mistakenly left a soda on top of one of the servers. Which item should you tend to first—and why? Which item would you address second—and why? Which item did you leave for last—and why? Document your findings and be prepared to share your thoughts with your class.

Color profile: Generic CMYK printer profile
Composite Default screen

Lab Projects

• Lab Project 20.1

Use the Internet to find Network+ practice questions. Try to locate as many troubleshooting scenario questions in the time allowed. Share your findings with classmates who have done the same. The more practice questions you cover as a group, the better prepared you will be handling real questions on the Network+ exam.

• Lab Project 20.2

Use the Internet to research the cost of four hardware tools used in troubleshooting mentioned in this chapter. Be sure to include shipping charges to your school's ZIP code. Use the following spreadsheet as a guide to tally your items.

Item: Model	**Web Site**	**Cost (Include S & H)**
Cable tester:		
		$
Hardware loopback device:		
		$
Protocol analyzer:		
		$
Toner:		
		$

ch20p26.prn
P:\010Comp\BasTch2C\560-9\ch20.vp
Wednesday, June 30, 2004 12:59:08 PM