

The Complete PC Tech

In this chapter, you will learn how to

- Describe how computers work
- Implement a troubleshooting methodology
- Describe a technician's toolkit

When a mission-critical computer goes down, regardless of the industry, people get upset. Workers can't work, so they feel guilty. Employers can't get product out on time, so they feel anxious. Supervisors blame employees for fouling things up, or at least the employees fear such blame, even if they did not break the machine.

Into this charged atmosphere comes the tech, ready to fix the computer and move on to the next challenge. Accomplishing this task, though, requires three things: First, a good tech must know the broken machine inside and out—how it's *supposed* to work when working properly. Second, the tech has to calm the workers and supervisors, and get answers to questions to gain relevant information about the problem. Third, the tech must troubleshoot the problem and fix the machine.

This chapter starts with an overview of how computers work and then dives into a section on dealing with customers and how to get them to tell you what you need to know and smile about it. The chapter wraps up with a proven troubleshooting methodology to help you figure out the source of problems and point you to the fix quickly.

Essentials

How Computers Work

You've spent a lot of time going through this book, reading about technologies and components in great detail. Each chapter contained troubleshooting information and methodologies for the components explained in that chapter. In Chapter 5, "Microprocessors," for example, you learned all about CPUs, from how they work to how to install them. You also learned about issues specific to CPUs, including the potentially difficult task of adding or removing the fan and heat sink assembly that all CPUs require. In Chapters 11, "Hard Drive Technologies," and 12, "Implementing Hard Drives," you dove into hard drives in gory detail. With each chapter,

you added more and more information about the pieces that make up the personal computer today.

In this chapter, I want you to distill that knowledge, to think about the computer as a coherent machine. Each of the computer's components works together to enable people to produce some amazing things.

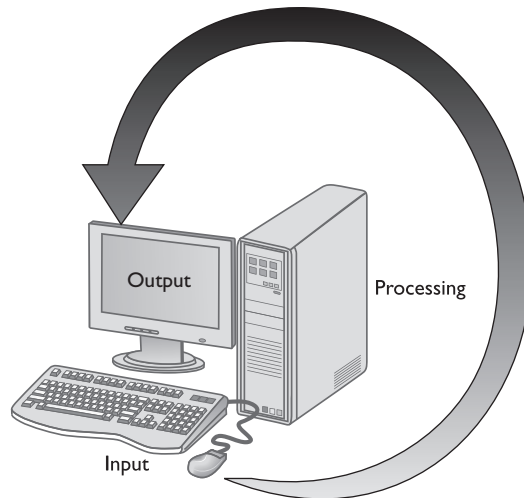
To master the art of troubleshooting as a PC tech, you need to approach a technical problem and answer one question: "What can it be? What can be causing this problem?" (Okay, that was two questions, but you get the idea.) Because every process involves multiple components, you must understand the interconnectedness of those components. If Jane can't print, for example, what could it be? Connectivity? Drivers? Paper jam? Slow network connection? Frozen application? Solar flares? Let's look at the process.

Way back in Chapter 3, "The Visible PC," you learned about the four parts of the computing process: input, processing, output, and storage. Let's take a moment to review the computing process, this time to see how you can use it to help you fix computers.

When you run a program, your computer goes through three of the four stages of the *computing process*: input, processing, and output (Figure 27-1). Input requires specific devices, such as the keyboard and mouse, that enable you to tell the computer to do something, such as open a program or type a word. The operating system (OS) provides an interface and tools so that the microprocessor and other chips can process your request. The image on the monitor or sound from the speakers effectively tells you that the computer has interpreted your command and spit out the result. The fourth stage, storage, comes into play when you want to save a document and when you first open programs and other files.

Figure 27-1

Input, processing,
and output



Making this process work, though, requires the complex interaction of many components, including multiple pieces of hardware and layers of software. As a tech, you need to understand all the components and how they work together so that when something doesn't work right, you can track down the source and fix it. A look at

a modern program reveals that even a seemingly simple action or change on the screen requires many things to happen within the computer.

Games such as Second Life (Figure 27-2) are huge, taking up multiple gigabytes of space on an Internet server. They simply won't fit into the RAM in most computers, so developers have figured out ways to minimize RAM usage.



Figure 27-2 Second Life

In Second Life, for example, you move through the online world in a series of more or less seamlessly connected areas. Crossing a bridge from one island to another triggers the game to quickly update the information you're about to see on the new island, so you won't be out of the action and the illusion of being in the game world remains intact. Here's what happens when you press the w key on your keyboard and your character steps across the invisible zone line.



NOTE Second Life is a massively multiplayer online role-playing game (MMORPG) that offers a unique twist on the genre. You can create just about anything you can imagine, as far as your time and talent can take you. Second Life has a functioning economy that spills out into the real world, meaning you can buy and sell things within the game and turn that into real U.S. dollars, although the more common scenario is to spend real money to get virtual possessions.

The keyboard controller reads the grid of your keyboard and, on discovering your input, sends the information to the CPU through the wires of the motherboard (Figure 27-3). The CPU understands the keyboard controller because of a small program that was loaded into RAM from the ROM BIOS on the motherboard when the PC booted up.

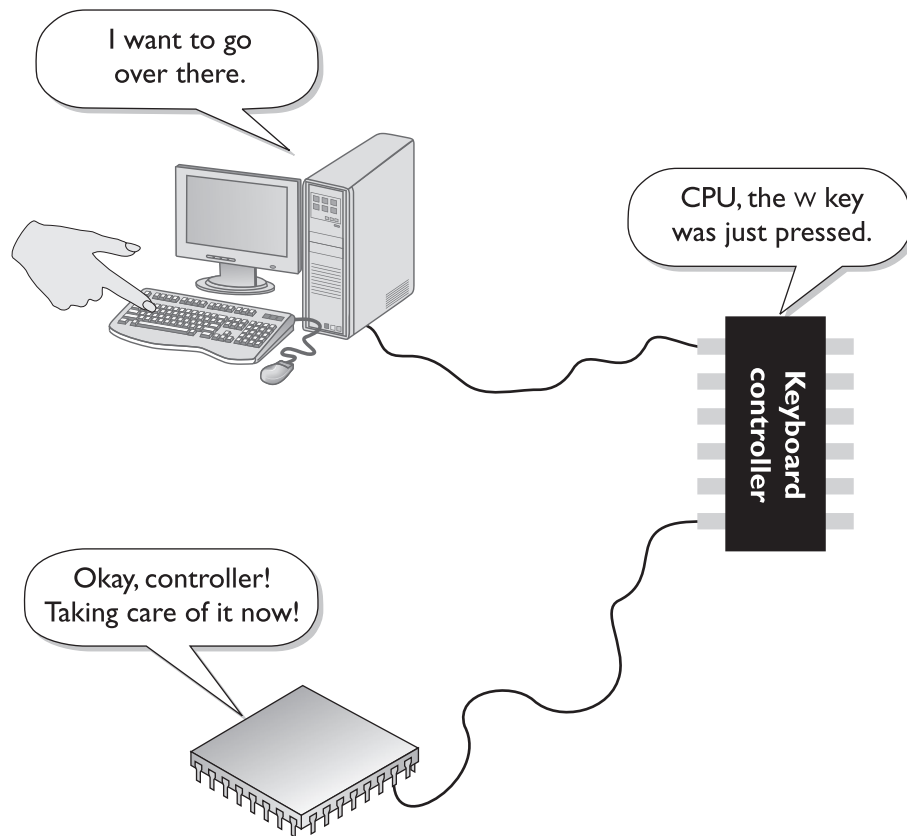


Figure 27-3 Keyboard to CPU

The CPU and the application determine what should happen in the game, and on discovering that your character is about to cross the zone line, they trigger a whole series of actions. The application sends the signal to the OS that it needs a specific area loaded into RAM. The OS sends a signal to the CPU that it needs data stored on the hard drive plus information stored on the Second Life servers. The CPU then sends the commands to the hard drive controller for it to grab the proper stored data and send it to RAM, while at the same time sending a command to the NIC to download the updated information (Figure 27-4).

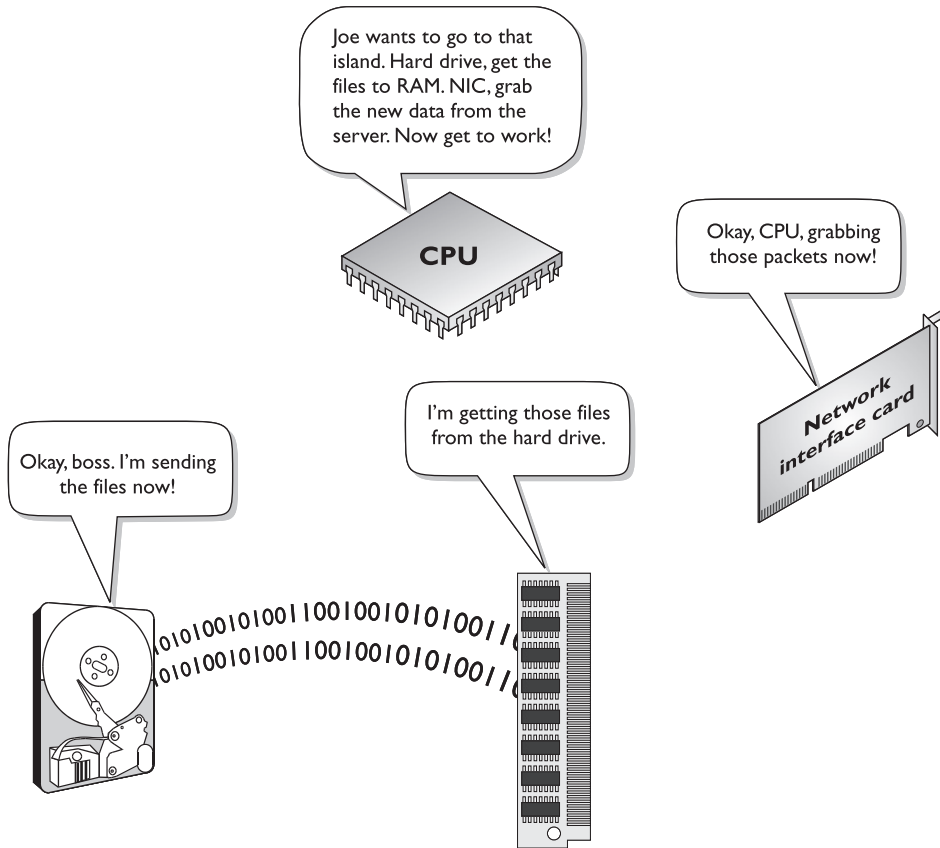


Figure 27-4 CPU to hard drive and NIC

The hard drive controller tells the hard drive to cough up the data—megabytes worth—and then sends that data through the motherboard to the memory controller, which puts it into RAM and communicates with the CPU when it's finished. The network card and network operating system communicate with the Second Life servers and download the necessary updated information. The CPU then uses the application and OS to process the new data, sending video data to the video card and sound data to the sound card, again through the wires on the motherboard (Figure 27-5).

The video card processor puts the incoming data into its RAM, processes the data, and then sends out commands to the monitor to update the screen. The sound card processor likewise processes the data and sends out commands to the speakers to play a new sound (Figure 27-6).

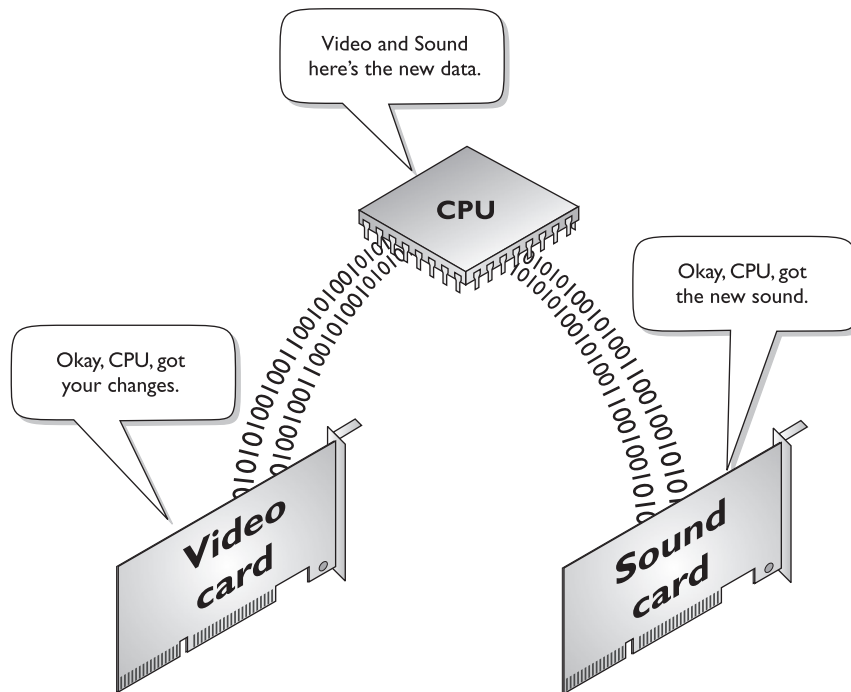


Figure 27-5 CPU to video card and sound card



NOTE Windows Vista has raised the bar on video demands in a big way, so the video card in your users' systems can make a remarkable difference in their experience. Vista uses the video card to produce many of the cool visual effects of the interface. This means that a low-end video card in an otherwise serviceable machine can cause Vista to misbehave. Unless your client is gaming, there's no reason to drop \$300+ on a video card, but assembling or recommending a system with yesterday's video is not necessarily a good thing!

For all of this to work, the PC has to have electricity, so the direct current (DC) provided by the power supply and the alternating current (AC) provided to the power supply must both be the proper voltage and amperage.

Finally, because Second Life is a network application, the OS has to send information through the NIC and onto the Internet to update everyone else's computer. That way, the other characters in the game world see you move forward a step (Figure 27-7).

What do you see or hear with all these electrons zipping all over the place? Out of a seemingly blank vista (Figure 27-8), a castle begins to appear, building itself piece by piece as your computer processes the new information and updates the video screen. You hear music begin to play from your speakers. Within a few seconds, with the data

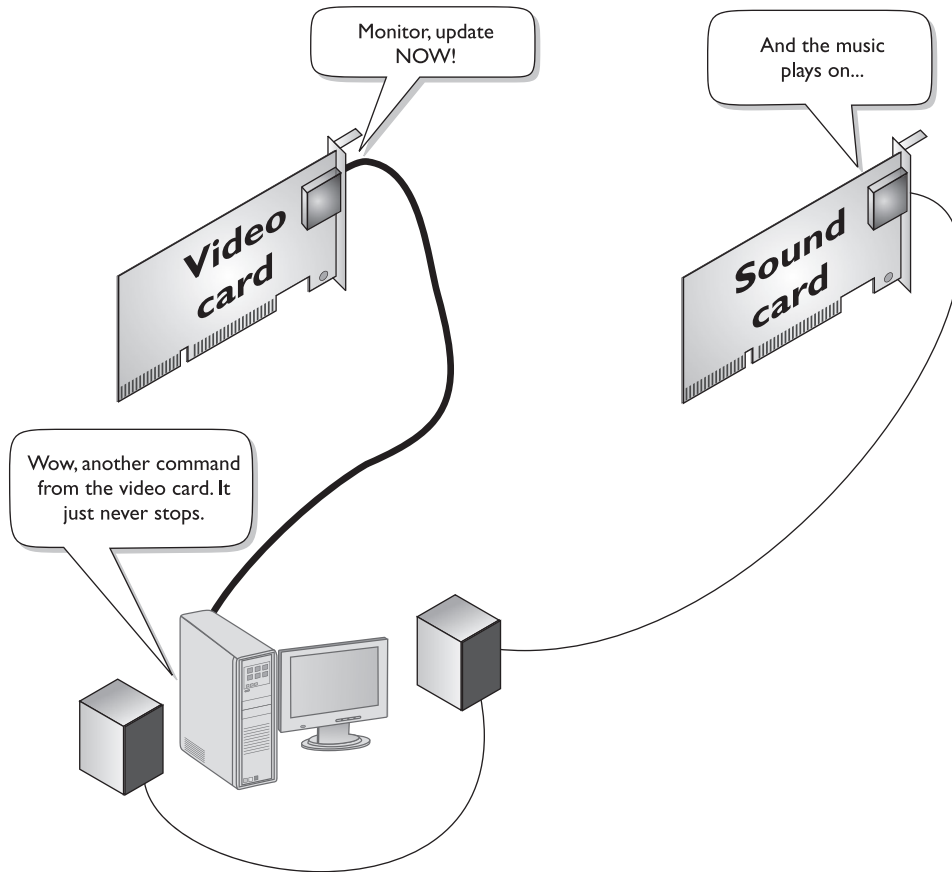


Figure 27-6 Updating the screen and speakers

describing the new island fully downloaded and processed, the world on your monitor looks very different (Figure 27-9). That's when all goes well. Many megabytes of data have flowed from your hard drive and across the Internet, been processed by multiple processors, and sent to the monitor and the speakers.

To keep the action continuous and unbroken, *Second Life*, like many current online games, uses a process of continuous or *stream loading*: your computer constantly downloads updated information and data from the *Second Life* servers, so the world you see changes with every step you take. When done right, stream loading can do some amazing things. In the GameCube game *Zelda*, for example, the game anticipates where you will go next and loads that new area into RAM before you take the step. You can be in one area and use a telescope to zoom in on another fully developed area (Figure 27-10), making the experience amazingly seamless, just like real life.

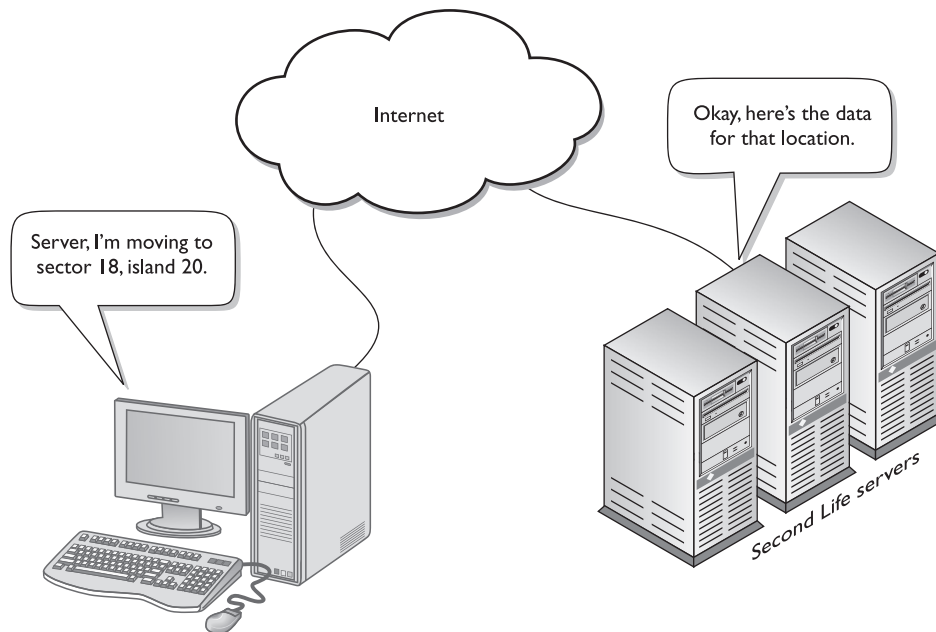


Figure 27-7 PC to Second Life servers



Figure 27-8 New area loading



Figure 27-9 Castle completed

Figure 27-10
Zelda zoomed



Anytime you have a problem on a computer that's not immediately and clearly obvious (such as not powering up due to not having a power cord plugged in), use should use the computing process to help you zero in on the problem. But the computing process only defines the pieces and their interaction – but good troubleshooting requires more than just rummaging around to find the problem. You need to combine the computing process with a troubleshooting theory to get the job done.



NOTE What applications do you use on your computer? What applications do your clients or potential clients use? Analyzing them in terms of the computing process can help elevate your troubleshooting game by a huge factor. Take for example Microsoft Excel, a program designed to let you take numbers and turn them into charts, among other things. When you sit down to work in Excel, you load it from the hard drive into memory and then put information in via the keyboard and format things with the mouse. Every click requires the CPU to analyze and update RAM and the video card information.

Troubleshooting Theory

Troubleshooting theory is nothing more than a set of mental steps you use along with the computing process to diagnose and fix a computer. Troubleshooting theory includes talking to users to determine how and when the problem took place, determining a cause, testing, verification, and documentation. Techs use a number of good troubleshooting theories. Luckily for those taking the CompTIA A+, CompTIA clearly defines their vision of troubleshooting theory in the 220-701 exam.

2.1 Given a scenario, explain the troubleshooting theory

- Identify the problem
 - Question the user and identify user changes to computer and perform backups before making changes
- Establish a theory of probable cause (question the obvious)
- Test the theory to determine cause
 - Once theory is confirmed determine next steps to resolve problem
 - If theory is not confirmed re-establish new theory or escalate
- Establish a plan of action to resolve the problem and implement the solution
- Verify full system functionality and if applicable implement preventative measures
- Document findings, actions and outcomes

Identify the Problem

There's a reason you're standing in front of a computer to repair it: something happened that the user of the computer has identified as "not good" and that's why you're here. The first step to *identifying any problem* is talking to the user. Get the user to show you what's not good. Is it an error code? Is something not accessible? Is a device not responding? Then ask the user that classic tech question (remember your communication skills here!): "Has anything recently changed on the computer that might have made this problem appear?" What you're really saying is: "Have you jacked with the computer? Did you install some evil program? Did you shove in a USB drive so hard you broke the connection?" Of course, you never say these things, simply ask nicely without accusing so the user can help you fix the problem (Figure 27-11).

Figure 27-11
Tech asking nicely



In most troubleshooting situations, it's important to back up critical files before making changes to a system. To some extent, this is a matter of proper ongoing maintenance, but if some important bit of data disappears and you don't have a backup, you know who the user will blame, don't you?



EXAM TIP The CompTIA A+ certification exams assume that all techs should back up systems *every time* before working on them, even though that's not how it works in the real world.

If you run into a partially functional system where you might have to reinstall the OS but can access the hard drive, you should definitely back up essential data, such as e-mail, browser favorites, important documents, and any data not stored on a regularly backed-up server. Because you can boot to a copy of Windows and go to the Recovery Console, you should never lose essential data, barring full-blown hard drive death.



NOTE Dead hard drives retain their data, so you can recover it—if you're willing to pay a lot of money. Having a good backup in place makes a lot more economic sense!

Establish a Theory of Probable Cause (Question the Obvious)

Now it's time to analyze the issue and come up with a theory as to what is wrong—a *theory of probable cause*. Personally, I prefer the word “guess” at this point because very few errors are so obvious that you'll know what to do. Fall back on your knowledge of the computing process to localize the issue based on the symptoms.

Keep your guesses...err...theories...simple. One of the great problems for techs is their desire to overlook the obvious problems in their desire to dig into the system (Figure 27-12).

Figure 27-12
Ford the Tech misses the obvious.



Outside the Case

Take a moment to look for clues before you open up the case. Most importantly, use all your senses in the process.

What do you see? Is a connector mangled, or a plastic part clearly damaged? Even if that connector or part works fine, the physical abuse could provide extra information. If the user can't connect to a network, check the cable. Was something rolled over it that could have broken the thin, internal wires? Is that a jelly smear near the jammed optical drive door? (No pun intended, really!) A visual examination of the external computer is important.

When you put your hand on the system unit, does it feel hot? Can you feel or hear the vibrations of the fans? If not, that would be a clue to an overheating or overheated computer. Modern computers can run when overly hot, but generally run very sluggishly. If you run through basic malware fixes, but a computer still runs poorly, think about excessive heat as a potential problem.

If you spend a moment listening to the PC, you might get some clues to problem sources. As you'll recall from Chapter 12, "Implementing Hard Drives," a properly running hard drive doesn't make a lot of sound; just a regular hum from the spinning platters. If you hear clicking or grinding sounds from a drive, that's a very bad sign and a very important clue! Excessive thrashing or disk access can likewise lead to some potential problem areas, such as insufficient RAM or a badly fragmented drive.

Finally, don't forget your nose. If you smell the unmistakable odor of ozone, you know that's the smell electronic components give off when they cook or are simply running much too hot.

Inside the Case

Use all your senses when you go inside the system unit as well. Do you see any physical damage? Check the motherboard capacitors if you have a dead PC. Properly working capacitors should be nice and (usually) blue; they definitely shouldn't look like partly melted batteries or be bulging at the seams. Fans should be spinning. The power supply shouldn't be blistering hot. You should be able to localize sounds better with the case off, and any smell of cooking components will definitely be stronger.

As you do your inspection, both outside and inside the case, don't jump to conclusions too quickly. When you see a problem that looks obvious, stop and give yourself an "Are you sure?" moment. Consider the possibility that a problem you've seen 50 times before might *not* be the same thing. Of course, it probably *is* the same problem, but that moment of consideration might save you trouble later.

Okay, so you've decided on a theory that makes sense. It's time to see if your theory is actually the problem at hand—give it a test to see if it fixes the problem.

Test the Theory to Determine Cause

The biggest challenge to fixing a computer is that the theory and the fix pretty much prove themselves at the same time. Let's go back to the situation where the power plug isn't in the wall. You observe that nothing happens when the PC power button is pressed. You check the back of the power supply and see that the power cable is plugged in. You then look at the wall outlet...not plugged in! So you quickly plug it in and *tada!* It works! You're a hero! In this simple case, your theory and your test are done in virtually the same move (Figure 27-13).

Of course, most problems aren't that easy. In many cases, *testing your theory* does nothing more than verify that something is broken. Let's say you install an update for your video driver. You reboot, log into Windows and suddenly the screen freaks out. Ah, a bad video driver, right? So to test, you reboot the computer and press F8 to get the boot options menu. You select VGA Mode (2000/XP) or Low Resolution Mode (Vista/7) and reboot. Now the computer boots up just fine. You know the problem,

Figure 27-13
Ford the Tech
is a hero!



but how do you fix it? Roll back the video driver? Reinstall an older driver? Try downloading another copy of the new driver and installing the video driver again? Any of these solutions may be the right one. Choose one and go for it (Figure 27-14).

Uh oh, your first guess was wrong: the video is still messed up. No worries, just try another one of your theories. In most cases you'll just pick another and try again. But sometimes there's a point where the problem is bigger than you. It might be a problem on a server that you're not authorized to configure. It might be a problem with a user account, and techs in your company aren't allowed to change user accounts. It might be a problem with an in-house program and you don't have the skills to fix it. In these cases, you must *escalate* the problem.

Escalation is the process your company (or sometimes just you) goes through when you—the person assigned to repair a problem—are not able to get the job done. It's okay to escalate because no one can fix every problem. All companies should have some form of escalation policy. It might mean calling your boss. It might mean filling out and sending some in-house form to another department. Escalation is sometimes

Figure 27-14
Ford the Tech
takes a chance!



a more casual process. You might want to start researching the problem online; you might want to refer to in-house documentation to see if this problem has appeared in the past. (See “Document Findings, Actions, and Outcomes” later in this chapter.) You may want to call a coworker to come check it out (Figure 27-15).

Verify and Prevent

Fantastic! Through either your careful work or escalation, you’ve solved the problem, or so you think. Remember two items here. First, even though you think it’s fixed, the user/customer might not think it’s fixed. Second, try to do something to prevent the problem from happening again in the future, if possible.

Figure 27-15

Ford the Tech asks for help from Scott.



Verify Full System Functionality *Verifying* full system functionality is CompTIA's way to tell you to make sure the user is happy. Let's say a user can't print. You determine that the printer spool is stalled due to a locked-up laser printer. You reset the printer and the jobs all start printing. Job done, right? Well, during your theory testing, you switched the default printer to another laser printer on the third floor. The user doesn't know how to set the default printer back to the one you just fixed.

The best way to verify full system functionality is to have the user do whatever she needs to do on the repaired system for a few minutes while you watch. Any minor errors (such as incorrect default printers) will quickly become apparent, and you might learn some interesting aspects of how the user does her job. Knowing what your users do is critical for good techs to help them do their jobs better (Figure 27-16).

Figure 27-16

Ford the Tech sticks around and watches.



If Applicable, Implement Preventive Measures A very smart tech once told me: “A truly good support tech’s work goal should be to never have to get out of his chair.” That’s a pretty tall order, but it makes sense to me. Do whatever you can to keep this problem from repeating. For some problems, there are obvious actions to take, such as making sure anti-malware is installed so a computer doesn’t get infected again. Sometimes there’s no action to take at all: nothing can prevent a hard drive that decides to die. But you can take one more critical action in almost every case: education. Take advantage of the time with the user to informally train him about the problem. Show him the dangers of malware or tell him that sometimes hard drives just die. The more your users know, the less time you’ll spend out of your chair.

Document Findings, Actions, and Outcomes

I think the famous historian George Santayana would have made a great PC technician when he said, “Those who cannot remember the past are condemned to repeat it.” As a tech, the last step of every troubleshooting job should be documentation. This documentation might be highly formalized in some organizations, or it might just be a few notes you jot down for your own use, but you must document! What was the problem? What did you do to fix it? What worked? What didn’t? The best guide to use for documentation is: “What would I have liked to have known about this problem before I walked up to it?” Good documentation is the strongest sign of a good PC tech (Figure 27-17).

Figure 27-17

Ford documents a successful fix.



Documenting problems helps you track the troubleshooting history of a machine over time, enabling you to make longer-term determinations about retiring it or changing out more parts. If you and fellow techs fix a specific problem with Mary's machine several times, for example, you might decide to swap out her whole system rather than fix it a fourth time.

Documenting helps fellow techs if they have to follow up on a task you didn't finish or troubleshoot a machine you've worked on previously. The reverse is also true. If you get a call about Frank's computer, for example, and check the records to find other service calls on his computer, you might find that the fix for a particular problem is already documented. This is especially true for user-generated problems. Having documentation of what you did also means you don't have to rely on your memory when your coworker asks what you did to fix the weird problem with Jane's computer a year ago!

Documenting also comes into play when you or a user has an accident onsite. If your colleague Joe drops a monitor on his foot and breaks both the monitor and his foot, for example, you need to fill out an *incident report*, just as you would with any kind of accident: electrical, chemical, or physical. An *incident report* should detail what happened and where it happened. This helps your supervisors take the appropriate actions quickly and efficiently.

Tech Toolkit

Way back in Chapter 2, "Operational Procedures," you learned the basic parts of a *tech toolkit*—a Phillips-head screwdriver and a few other useful tools, such as a Torx wrench and a pair of tweezers. Any good tech makes a point to carry around at least these tools (Figure 27-18). Over time, you'll add more tools as your experience grows. But your toolkit won't stop at Phillips screwdrivers and Torx wrenches. You should include two other types of tools in your tech toolkit: utilities and field replaceable units (FRUs).

Figure 27-18
Typical technician
toolkit



Utilities

Windows comes with plenty of handy utilities, but there are times when you need to use stronger tools than these. The PC industry has thousands of third-party utilities that techs might use to diagnose and repair PCs. Sadly, there is no single selection of tools I can tell you to get: the tools one experienced tech uses differ from the tools I use based

on experience, skills, and job functions. That doesn't mean you haven't been hearing my opinions! Throughout this book you've seen quite a few third-party utilities, and many of them are on the CD that comes with your book for you to use right away. Given that you already have my opinion on so many tools, let's instead talk about the types of third-party tools you'll find in most tech toolkits.



NOTE The Tech Forums on the Total Seminars Web site (www.totalsem.com/forums) has an entire section dedicated to nothing more than what we call "Cool Tools." If you want an opinion on hundreds of tech utilities, come over to the Tech Forums and see what we like...and what we dislike.

Many techs like to challenge me on the idea of carrying around tools. They say, "I know what I like; I'll just download them!" Well, sometimes you don't have Internet access and sometimes you need tools ready to go (see Malware Cleaners). Granted, there are a number of tools that I would never carry on me. For example, device drivers change so often that keeping a copy on disk is a waste of time. The tools I'm listing here are the ones I promise you want on disk, ready to go when you run into trouble.

Malware Cleaners

Ask any tech who works for Best Buy's Geek Squad: "What do you spend the most time doing on jobs?" and they will reply: "Cleaning malware." The first tech utilities you want in your toolkit are malware cleaners. To truly do it right, put your malware cleaners on a bootable optical disc because malware can sometimes attack a malware cleanup tool as you install. If you're not sure what to try first, grab a copy of the Ultimate Boot CD and try some of their fine cleanup tools.

Anti-Malware

After you clean the machine, make sure you have a copy of an anti-malware program that you can install so the user's machine doesn't get corrupted again. We talked about some of the popular freeware versions, but plenty of techs strongly prefer the pay versions and keep copies to sell to their customers.

Boot Tools

You need a tool that's better than the System Configuration utility at controlling what programs autostart on your Windows PC. OK, in this case, one tool stands out so strongly that you have to get it: Mark Russinovich's superb AUTORUNS. It's vastly superior to the built-in System Configuration utility and gives you incredible control of everything that's autostarting. Sure there are others, but I love AUTORUNS. Get it at www.sysinternals.com.

Password Clearer

If you lose a Windows password, especially the Administrator password, and you can't log on to a system, you're in trouble. A number of tools let you reset any user's passwords. These programs don't let you *see* a password, only reset them, so if you have a user with encrypted folders, those won't be recoverable. None of these programs are easy to use, but when the alternative is not logging in, they will save you. Pretty much all of these programs run from a boot CD.

ZIP File Tool

Windows can read many compressed file formats (such as ZIP), but so many others are in use that it's often a good idea to have your own copy handy. I like 7-Zip (www.7-zip.org/), but there are plenty of equally popular alternatives.

Backup

Odds are good that from time to time you'll need to do a backup for a system. I keep an external hard drive with a backup tool so I can go in fast and make backups of the most critical files.

Don't Forget Your Thumb Drives!

I guess I'm showing my age when I talk about putting all these tools on CD media. Sorry, I'm old. Almost all of these tools work equally well on USB thumb drives. I keep my tools on CD media because I still run into the occasional system that doesn't have CMOS settings for a bootable thumb drive.

Field Replaceable Units (FRUs)

Always carry several *field replaceable units (FRUs)*—a fancy way to say *spare parts*—when going to a job site or workstation. Having several known good components on hand enables you to swap out a potentially bad piece of hardware to see if that's the problem. Different technicians will have different FRUs. A printer specialist might carry a number of fusers, for example. Your employer will also have a big effect on what is an FRU and what is not. I generally carry a couple of RAM sticks (DDR, DDR2, and DDR3), a PCI video card, a NIC, and a 300-watt power supply.



EXAM TIP The CompTIA A+ exams expect you to know that FRUs are computer components such as cards, memory, hard drives, processors, or power supplies that can be easily replaced in the field.

I keep a PCI video card in my kit, by the way, because every computer made in the past ten years has PCI slots. If you run into a system with video problems, you can almost always simply slap in the PCI video card and discover quickly whether the AGP or PCIe video card or card slot is a problem. If the computer boots up with the PCI video card but fails on the PCIe card, for example, you know that either the PCIe card or the slot is causing the problem.

Chapter Review Questions

1. Once you have assessed the problem with the user's machine, it is important to do what before making changes?
 - A. Attach your grounding strap
 - B. Nothing; start fixing the problem immediately
 - C. Back up critical files
 - D. Get a signed release from the user

2. While working at the help desk, you get a call from a distraught user who says she has a blank screen. What would be useful follow-up questions? (Select two.)
 - A. Is the computer turned on?
 - B. Is the monitor turned on?
 - C. Did you reboot?
 - D. What did you do?
3. Computers are complex machines, but many times the answer to a problem is:
 - A. Obscure
 - B. Complex
 - C. Obvious
 - D. Very hard
4. What tool should be in every technician's toolkit?
 - A. Pliers
 - B. Hammer
 - C. Flat-head screwdriver
 - D. Phillips-head screwdriver
5. According to troubleshooting theory, the first thing you should do when you arrive to work on a computer is:
 - A. Identify the problem.
 - B. Ask how the user broke the machine.
 - C. Start working on the problem immediately.
 - D. Backup critical data.
6. Once you have ascertained the computer's problem and backed up the critical data, what should you do?
 - A. Establish a theory of probable cause
 - B. Start fixing the machine
 - C. Question users more to find out how they caused the problem
 - D. Document
7. Once you have developed a theory of what has caused the problem, then what should you do?
 - A. Prepare the machine to be taken back to the workbench.
 - B. Accuse the user of being an idiot for causing the problem.
 - C. Test the theory to determine cause.
 - D. Escalate the problem.

8. After successfully repairing a machine, it is vitally important to:
 - A. Do nothing; your job is done.
 - B. Admonish the user for causing so much work for the IT department.
 - C. Document your findings.
 - D. Lock it down so the user can't cause the same problem again.
9. You've just installed new printer drivers into Roland's computer for the big networked laser printer. What should you do to complete the assignment?
 - A. Document that you installed new printer drivers.
 - B. Tell Roland to print a test page.
 - C. Print a test page and go to the printer to verify the results.
 - D. Print a test page and go to the printer to verify the results. Document that you installed new printer drivers successfully.
10. What's an FRU?
 - A. Foreign replacement unit—a cheaper part to replace an expensive American-made part
 - B. Field replacement unit—a known good computer part to swap with a suspect part as part of the troubleshooting process
 - C. Free repair unit—a gimmick used by some companies to provide free tech support for the first-time customer
 - D. Short for FRU Linux, a Linux distribution

Answers

1. C. You should never lose data if you have an opportunity to back it up.
2. A, B. Go for the simple answer first. When faced with a blank screen, check to see if the computer and the monitor are on.
3. C. Don't overlook the obvious answer such as an unplugged power cord.
4. D. Every tech's toolkit should have a Phillips-head screwdriver, at the very least.
5. A. The first thing you should do is identify the problem.
6. A. You should establish a theory of probable cause once you have ascertained the problem and backed up data.
7. C. Once you think you know the cause, test your theory.
8. C. At the end of a repair you should always document your findings.
9. D. Always verify the results once you finish troubleshooting. Then document what you accomplished.
10. B. A field replacement unit is a known good computer part.